TMF814 Network Simulator

A standalone, Java-based network simulator for testing communication over the network management protocol TMF814.

Master of Science Thesis

LOUISA LUCIANI
MIKAEL RIEDEL

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, September 2010
The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

TMF814 Network Simulator
A standalone simulator for testing communication over the network management protocol TMF814.

LOUISA LUCIANI
MIKAEL RIEDEL

© LOUISA LUCIANI, September 2010.
© MIKAEL RIEDEL, September 2010.

Examiner: Sven-Arne Andreasson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden September 2010
Abstract

This thesis covers the process of developing a network management system simulator with support for northbound communication over the CORBA-based protocol TMF814. The simulator allows the user to import real data from an existing network management system. The user can view and configure the data through a GUI. The simulator acts as a TMF814 server for clients on higher management network levels. Much of the focus in this project lies on the architecture’s modularity, which can be divided into three areas: a graphical user interface, a northbound interface and a database interface.

The intention of the final product is to facilitate TMF814 integrations, but parts of the code can be re-utilized for multiple purposes. Modules can be attached and extracted to make custom solutions that require either processing or visualization of network management data or simply TMF814 server functionality. Developing the simulator has lead to a deeper understanding of the protocol and its underlying structures. Therefore, experiences and pitfalls that might be useful for the prospective TMF814 developer are shared.

As the comprehensiveness of the protocol revealed itself during the course of the project, we have a final deliverable with a small portion of the server side functionality implemented. Apart from the server implementation, the product parses client data and visualizes it graphically. At the end, we have successfully built a modular and maintainable platform, although more testing is necessary before we can verify correct behavior. We are also left questioning interoperability, abstraction, maintainability and legal issues of the protocol’s underlying infrastructure.

Keywords: Fault Management, OSS, Operation Support Systems, Network Supervision, Network Management
Preface

This document contains a master’s thesis for the Department of Computer Science and Engineering at Chalmers University of Technology. The thesis will be conducted in cooperation with Global Service Delivery Center (GSDC), Ericsson. We will specifically be working with GSDC OSS integration, a team in Gothenburg, Sweden that provides integration solutions, customizations and support of Operations Support Systems.
**Vocabulary**

AM - Accounting Management.

CM - Configuration Management.

**CORBA** - Common Object Request Broker Architecture.

DBI - Database Interface.

EMS - Element Management System.

EOS - Ericsson OSS Simulator, this is the product developed during this project. Sometimes referred to as the simulator.

**FCAPS** - Fault, Configuration, Accounting, Performance, Security.

FM - Fault Management

ME - Managed Element, network element in TMF814.

**MLSN** - MultiLayerSubnetwork, subnetworks in TMF814.

MVC - Model View Controller.

NBI - North Bound Interface.

NMS - Network Management System.

**Northbound** - Communication to overlaying systems.

OMG - Object Management Group, founder of CORBA.

OSS - Operations Support System.

PM - Performance Management.

SBI - South Bound Interface.

SM - Security Management.

**Southbound** - Communication to underlying systems or equipment.

TL - Topological Link, a network link.

TMF - TeleManagement Forum.
# Contents

1 Introduction

1.1 Background ........................................... 1
1.2 Problem description ................................. 3
1.3 Purpose .................................................. 4
1.4 Objective .............................................. 4
1.5 Scope .................................................... 5
1.5.1 Demarcations ....................................... 5

2 Research

2.1 Network Management ................................. 6
2.2 ServiceOn ............................................ 8
2.3 CORBA ................................................ 9
2.3.1 Overview of the architecture ..................... 9
2.3.2 IDL .................................................. 10
2.3.3 GIOP ............................................... 10
2.3.4 Object Services ..................................... 11
2.3.5 Competing Technologies ......................... 13
2.3.6 Alternative ORBs ................................ 14
2.4 TMF814 ............................................... 14
2.4.1 Structure ......................................... 15
2.4.2 Entry-point when implementing .................. 17

3 Method .................................................... 20

4 Design and Implementation ......................... 22

4.1 Data ................................................... 23
1 Introduction

Mobile and fixed broadband access is increasing rapidly throughout the world. End users demand the ability to stay connected to anyone, at any time, regardless of location with high speeds and dependability. Consequently, the providers of network-infrastructure must be able to manage and supervise their equipment in an efficient way. Operational Support Systems (OSS) provide tools for managing networks and network elements. Managing a network includes monitoring the network to detect problems, retrieving performance and inventory data, and configuring the resources.

1.1 Background

The communication between the OSS and the network elements, ie. requesting data, is called Southbound Communication. Today, Network Management Systems are often distributed, so they must interact with each other and be managed by higher level systems. The communication to the higher level systems is called Northbound Communication. In 2005, Ericsson acquired a majority of Marconi Corporation [1], who's key assets include a tool suite called ServiceOn that provides management tools on both a network- and element-layer for different types of networks including optical and microwave [2]. It has multiple interfaces for both southbound and northbound communication. For northbound communication, TMF814 is one of the most promising protocols. TMF814 is a CORBA-based protocol specifically designed for network management.

To begin looking at the background of TMF814, one must look at the history of it’s underlying architecture, CORBA. Originally, there were remote procedure calls (RPC) to handle distributed communication. In the early 90’s, much of the existing middleware was tied to specific languages and platforms, and more adaptability and functionality was required as distributed computing became more prevalent [3]. Creating communication between heterogeneous environments was very difficult. A middleware called Common Object Request Broker (CORBA) was developed by Object Management Group (OMG) in 1991 [4]. CORBA was a distributed architecture that made it possible to design and implement a distributed object oriented system as a group of modular components where complexity could be hidden behind layers of abstraction. Initially, only a mapping to C was provided.

In 1997, CORBA 2.0 was released with a C++ mapping, followed by a Java mapping in 1999 [4]. Around this time, CORBA gained a sufficient amount of popularity and a standardized protocol was provided. During CORBA’s growth, developers began criticizing the complexity of creating CORBA applications. This was mainly due to the inconsistent, comprehensive and complex API. The platform had a steep learning curve and caused long development processes and high fault rates, which was too expensive for many companies. Microsoft chose to compete with CORBA by producing DCOM (Distributed Component Object Model), but
DCOM was not a big success either as it was platform dependent. Eventually DCOM lost popularity and Microsoft returned with .NET remoting, and also the publication of SOAP, which used the popular XML as the encoding for remote function calls [5]. This caused some fallbacks for OMG. Today CORBA is growing in the real time and embedded systems development market, and is commonly used for communication between components within a company’s network. In the mean time XML, SOAP, Web services, and service-oriented architectures are gaining popularity.

In history, you often see large corporations cooperating on standards to promote efficiency, a scenario that benefits all. In 1988, OSI/Network Management Forum was founded by eight leading telecom companies of the time to advance the availability of network management products [6]. In 1998, They changed their name to TeleManagement Forum, and began a commitment to OSS standards. Today, members include people from 700 different companies in 185 different countries. Although many solutions are still built upon proprietary and custom technologies, TM Forum is regarded as the most authoritative source for standards and frameworks in OSS.
1.2 Problem description

Today ServiceOn allows the user to create virtual nodes. However, it can only communicate northbound over TMF814 if the node is physical. The GSDC OSS-integrations team works with customizations, integration services and solutions design of the product and often, a solution specific script will be created or an adjustment in the system will be made for a service provider. In order to verify the expected outcome, tests need to be performed on a physical network. Using physical network elements for testing is considered too expensive, so GSDC often performs the tests on-site prior to the installation. Because the clients are located across the world and the team often has a very small time-limit to assure the promised functionality, they can not afford unexpected results.

Not only is this process costly and inefficient, but visiting a client and not being able to proceed with the integration is both a loss for Ericsson and for the clients. Also, different network elements differ in functionality, for instance, in the types of alarms they can send and in their configuration possibilities. Because the virtual nodes in ServiceOn are not perfect representations of physical network elements, they can not be used for testing new integrations. This is why GSDC has expressed a need for a virtual network simulator that communicates TMF814. The desired outcome is a standalone application that simulates an optical network and a manager on the Network- and Element- Management layer, depicted in Figure 1. The simulator should be able to communicate with higher management layers over TMF814.

Figure 1: What the simulator should implement in correlation to the management layers. (EOS is the name of the simulator)
1.3 Purpose

The purpose of this thesis from an academic perspective is to develop a deeper understanding of modular architectures using principles from object oriented design as well as gaining some insight into CORBA and the protocol TMF814. This thesis will also cover our design process and experience upon building a large architecture with multiple technologies. Knowledge will be gained throughout the entire life cycle of the product development, from idea to requirement specification to implementation and distribution of the simulator. Although efficiency and usability are important, the focus will mostly lie on the stability and modularity of the implementation. This is because several subcomponents within this project, such as parsing a data from a client database, might be useful to extract and use in another context.

1.4 Objective

In this project, a simulator for a network elements and a northbound interface will be developed that complies with TMF814. The application must work on Windows Vista OS.

Modularity Because of the large scale and potential of the project, our primary goal is to build an architecture that is scalable. Modularity is a fundamental part of object oriented design and in our case it is not difficult to see the need for decomposition of highly cohesive but loosely coupled modules.

Abstraction The functionality of a module should be characterized by a contractual interface that captures the behavior of the module and allows polymorphism, to facilitate creation of new modules. Encapsulation will allow the implementation to be modified easily without affecting other parts of the system.

Performance The simulator should be lightweight and perform well, as it will mostly be used on single and dual-core hardware.

Value Because the simulator will be used for testing purposes, the value of the simulator relies heavily on its similarity to real networks. The only way to assure similarity to real networks and catch all unexpected cases is to actually read real network data. Our implementation will therefore include an interface for this. To increase reliability, we will perform tests and quality assurance reviews.

Usability The information provided by the simulator must be easily accessible through a graphical user interface with a high level of usability.

Maintenance Our goal is to develop code that is easy to understand and manage post-delivery, to allow for adjustment of the system according to evolving needs. We will therefore provide clean code and documentation.
1.5 Scope

TMF814 contains 456 functions divided into 15 packages (also called managers). Because of the comprehensive amount of functions available for TMF814, the scope of the project will have to be limited a certain amount of functionality. Because simulation of fault management is a priority for GSDC, a partial implementation of FM is considered a mandatory part of the feature-complete beta. The second priority is to include some inventory management and configuration functionality which will make the simulator more useful. Specifically, the Ems, ManagedElement and Equipment Manager packages will be in focus.

An interface for connecting more managers will also be developed, so that, for example, the FlowDomain or PerformanceManagement manager could be developed and easily integrated in the future. The simulator will include a GUI that displays the content of the simulator in a clear and useful way. It should be possible for the user to configure the parameters of the network elements, to send alarms and modify the setup of elements and links that are being simulated. The simulator should be able to populate its own database with information from other databases, e.g. from a service provider. This should be done through an interface developed during this project.

1.5.1 Demarcations

This project will not include full support for the entire protocol. Within the time frame that is given for this project it is not reasonable to implement the entire functionality. Also, the protocol is comprehensive and covers much more than Ericsson requires at the moment. In order to avoid a death march work schedule, the focus will lie on the previously mentioned managers (Ems, ManagedElement and Equipment Manager), with agility in mind. These demarcations reflect on the telecommunication areas as follows:

- FM, Fault Management will be covered, this is the highest priority for Ericsson
- PM, Performance Management will not be implemented during this project, but could be added in the future if a module is developed using the interface.
- IM, Inventory Management will only partly be implemented
- CM, Configuration Management will only partly be implemented

The simulator should only be able to communicate northbound, in other words, not southbound to actual network elements. The project doesn’t include mapping all the different types of databases the different service providers might use. However, it will include an interface so that it is possible in the future to build an adapter for an alternative database structure.
2 Research

In the following section we will briefly cover the concept of network management, to create a broader understanding of what the final product in this project is intended to be used for. In order to create system requirements and understand the needs of the OSS integrations team, an analysis of the current product portfolio must be done, which is what section 2.2 will cover. Section 2.3 will cover the distributed architecture used in this project - CORBA. Finally, the last section will describe the structure of the protocol TMF814.

2.1 Network Management

Telecommunication providers require Operational Support Systems to manage inventory of their network equipment, for configuring the components and for managing faults in the network. Effective management of a network infrastructure is necessary to ensure the quality of their services. Part of Ericsson’s Solutions portfolio consists of Network Management solutions for telecommunication operators that need to improve operation efficiency. Network management refers to the administration and maintenance of network elements in a system, and is commonly divided into 5 categories known as FCAPS - Fault, Configuration, Accounting, Performance and Security defined by Open Systems Interconnection(OSI).

At Ericsson, the solutions provided are separated into the following categories:

- Fault Management
- Performance Management
- Inventory Management
- Trouble Management
- Revenue Management

All these solutions can be integrated with each other in a single custom solution for a client.

Fault management acts upon error detection notifications by tracing and identifying faults, along with information such as their probable causes and their severity rate. When an alarm is triggered a notification is sent northbound to the overlying system that can be monitored by a system operator.

The solution provided for Performance Management collects, processes, and presents performance data in reports. This is interesting for clients because measurements of network capacity and usage statistics can help the provider maximize network performance to ensure better availability, reliability and quality.

The solution for Inventory Management displays information of the entire network infrastructure in a single framework. It allows the provider to configure the
inventory, to report functions to users, to perform process modeling and to create processes. Operators require inventory management to take care of operational issues and spare parts management, so that they can have an accurate overview of the network.

Trouble management solutions provide a set of tools that help the provider identify areas for improvement and facilitate taking the right steps to achieve operational efficiency. Trouble management supports Trouble tickets (a notification that something is wrong), change requests (CM and planned action), work orders, performance indicator reports and escalation of service level agreements (SLAs).

To take a step deeper into how these services are implemented, one must look at the access methods and standards used for the communication of these services. The protocols used for network management generally have to support multi-vendor, multi-technology communication. Several access methods support network device management, with some of the most common methods being SNMP, Windows Management Instrumentation (WMI), CMIP and CORBA. The type of access method used depends on the type of information the needs to be transferred, security, reliability, speed and other priorities that the service requester has.
2.2 ServiceOn

ServiceOn is a portfolio of Element, Network and Service Management tools developed by Ericsson (formerly Marconi, acquired 2005). It is a modular platform that provides functionality according to the recommendations from the FCAPS model [2]. ServiceOn can also be used to mediate data to a client’s custom Network Management System. ServiceOn has the ability to manage a broad mix of Ericsson products in many areas, including:

- Broadband Access
- Microwave
- Optical
- Metro Ethernet

In addition to these modules ServiceOn Element Manager can manage 3rd party SNMP based equipment. For communication with the network elements several protocols can be used. For instance, ServiceOn is capable of communicating over SNMP, TCP/IP and CLNS with network elements. All known network elements are stored in a database. As for northbound communication (or middleware), CORBA is used. In other words, the communication between the Element Management Layer the Network Management Layer and the Higher Management Layer uses CORBA based protocols. Figure 2 outlines the communication between the different Management layers.

![Figure 2: ServiceOn System Platform](image)
2.3 CORBA

2.3.1 Overview of the architecture

CORBA is a middleware architecture designed by OMG used to ensure language and platform-independent interoperability between a server’s objects and a client’s method calls. CORBA applications allow communication within the same address space as well as between remote address spaces. It follows the object-based model as opposed to a pure client/server model or service model, by providing the client with an encapsulating object-oriented interface. The object oriented approach is good for protocols where the client and server know exactly what to expect of each other (strictly defined in the protocol), so interfaces can simplify the client and server development.

Figure 3 illustrates how a Server and a client communicate over an Object Request Broker (ORB). The Stub and the Skeleton serve as proxies for clients and servers respectively. The ORB is the middleware that handles communication details between distributed objects. The ORB handles:

- Finding the object implementation for the request
- Preparing the object implementation to receive the request
- Communicate the data making up the request

![Diagram](image)

Figure 3: IDL files produce interfaces towards the ORB.

A request is issued using an object reference, an operation name and a set of parameters. This request is performed within the application code using IDL stubs as an interface. When the stub routine is called the object reference for the target object is mapped to the object reference as represented by the ORB.
2.3.2 IDL

CORBA uses an interface definition language (OMG IDL) to specify interfaces. An interface includes operations that servers promise to perform on behalf of clients. There is a specified mapping from OMG IDL to most of the popular programming languages, including Java. The transformation between CORBA IDL definitions and the target programming language is automated by a CORBA IDL compiler. These are the specific files that the compiler creates:

POA file. The POA file creates a piece of the ORB that manages the server-side resources. Acronym for Portable Object Adapter. The POA file is also called the server skeleton. A local servant extends this class and implements the functionality. The POA can activate and deactivate the servant depending on its usage.

Stub file. The client stub implements the Java interface.

Java interface. Java version of the IDL interface.

Helper file. Responsible for reading and writing the data type to CORBA streams.

Holder file. Delegates to the methods in Helper for reading and writing.

Operations file. Contains the methods from the IDL file, which is shared by both the stubs and the skeletons.

The POA file’s servant is responsible for executing the object’s operation. It can be activated and deactivated. The servant is encapsulated behind the interface and thus visible only to the server.

CORBA also provides an alternative method of invocation, namely Dynamic Invocation Interfaces (DII) [7]. DII provides an interface that allows requests to be dynamically built (in other words at invocation time, but not at compile time). DII was designed to enable requests to be issued by programs that were created before the interfaces for the operations were designed. The server side’s analogue to the client side’s DII is called DSI. Just like DII, the DSI allows an ORB to deliver requests to an object implementation that does not have compile-time knowledge of the type of the object it is implementing. The client making the request has no idea whether the implementation is using the type-specific IDL skeletons or is using the dynamic skeletons.

2.3.3 GIOP

In the OSI model, CORBA lies below a vendor-produced protocol (TMF814 in our case), and above GIOP (General Inter-ORB Protocol), which in turn is above the transport layer. Upon the request, the client uses the IDL stubs in the same way that it would have used a call to a library function. When the stub routine is
called, the target object is mapped to the object references represented by the ORB. The ORB is then responsible for locating the object implementation and routing the request to that implementation, as well as returning any results back. GIOP is an abstract protocol that allows for ORB interoperability, and can run on top of virtually all transport-layer protocols. The Internet Inter-ORB protocol (IIOP) is essentially GIOP that runs over TCP/IP. IIOP is incorporated in CORBA and defined as the standard, so it will be the protocol rather than GIOP to look more closely at.

IIOP defines a set of data formatting rules, called Common Data Representation (CDR). Objects publish identities and locations through Object References, which in IIOP are called Interoperable Object References (IOR). When a client wants to access a CORBA object, it first obtains an IOR for that object. Using the IOR, the client can then invoke methods on the object.

2.3.4 Object Services

Because of OMGs approach to keep components modular, some functionality that is needed for building applications upon CORBA require domain-independent interfaces, which OMG calls Object Services. Object Services include Naming, Event Notification, Life Cycle and Persistence Services [7].

- Naming Services allow a name to be bound to an object within a certain naming context.
- Event/Notification Services provide a facility for delivering event information.
- Life Cycle Services include creation, deletion, copying and moving of objects.
- Persistence Services provides management of CORBA object’s persistent state.

The Naming Service acts as the first connection point when a client wants to connect to a server. Since the IOR that is generated looks different each time the server is started, it is important that the client has the newest version each time it tries to establish a connection. The Naming Service provides the newest version of the server’s IOR-string and can give it to all clients that want to connect to the server. If a server is restarted and thereby a new IOR-string is required the server simply registers the new IOR-string at the NameService. The Naming Service then replaces the old string with the new one. In this way the clients easily can access the server with the latest version of the IOR-string.

To be able to handle many different servers on one single Naming Service, a tree structure is used to guide the client to the objects that are relevant. This structure resembles the structure of a file system. Naming Service employs a tree structure of Naming Contexts (see Figure 4). Naming Contexts have a name of their own and resemble directories in a file system, where they can be nested within each other. Naming Contexts specifically employ name-value tuples, where names are
structures with an identifier attribute and a kind attribute and values are object references. Names must be unique within a Naming Context (just like files within a directory). Object references can exist directly in the root of the tree or within any of the underlying Naming Contexts. The binding between a name and an object is called a name binding. The server implementation is responsible for building this tree. The binding of a name to an object is done using the bind operation defined in the NamingContext Interface. To be able to resolve a name the client must know the path of Naming Contexts and the final name it wants to resolve. The Objects are also referred to as bindings, because they contain the IOR-string that is bound to the actual server object. When resolving the IOR-string, the client can create references to the server through the orb.

![Diagram of tree structure produced in Name Service.](image)

Figure 4: An illustration of the tree structure produced in Name Service.

Notification Services is an extension to Event Services, and is specifically designed for decoupled messaging. For example, the parties involved do not have to know about each other. One can simply subscribe to a channel that another party broadcasts upon.

Notification Services defines three roles; the supplier, the event channel and the consumer. The supplier pushes event data to the channel and the channel pushes the data to consumers. The channel can also pull data from the supplier and the consumer can pull event data from the event channel. The push and pull functionality is built into the EventComm Module. The EventComm module defines four interfaces. The interfaces PushConsumer, PushSupplier, PullConsumer and PullSupplier. These interfaces include push or pull/try_pull as well as a disconnect
function. An EventChannelFactory object is used to return an object reference that supports the eventChannel interface, which in turn defines the administrative operations:

- ConsumerAdmin: a factory for adding consumers
- SupplierAdmin: a factory for adding suppliers
- An operation for destroying the channel.

The Event Notification Service specifies interfaces, and not implementation details. This is necessary to allow different implementations to provide different qualities of service (QoS).

### 2.3.5 Competing Technologies

There are three common types of middleware [8):

- Remote Procedure Calls (RPC)
- Message Oriented Middleware (MOM)
- Object Request Brokers (ORB)

In Remote Procedure Calls the client makes calls to procedures running on remote systems. In message oriented middleware, messages are sent to the client which are stored until they are acted upon. Object Request middleware makes it possible for applications to send objects and request services in an object-oriented system. When Remote procedure calls use object oriented principles, the term Remote Invocation or Remote Method Invocation is used.

Although web services are gaining popularity in the distributed technology market, object-based distributed technologies have the niche of being designed for use within an organization or a small number of collaborating organizations. Before a client can communicate with a server, object based systems must know the initial naming context, but they generally have the advantage of low verbosity and superior performance, both in bandwidth and processor usage compared to service-based systems.

If we look at the distributed object-based technologies, three of the most dominant competitors are:

**CORBA - Common Object Request Broker Architecture** CORBA is the most widely used standard in the non-Windows Market.

**.NET remoting** is a standard developed by Microsoft

**Java RMI - Remote Method Invocation Protocol** is a standard developed by Oracle (formerly Sun Microsystems, acquired 2009)
What the three have in common is that they provide an abstraction of complex networking implementations so that the developer can concentrate on business logic in an object oriented manner. They all provide interfaces that marshal the parameters and send them through a wire protocol to a remote system where an interface resembles the marshaled parameters and calls a function that returns an object.

The three technologies differ in the way they handle invocation of objects, garbage collection, security object identification and many other areas. They each have a unique network protocol and their own set of component models. The most relevant differences though are interoperability issues. While .NET Remoting can be used for cross-platform communication, but is optimized for communication between .NET based applications [5]. Furthermore Java RMI, only complies with servers and clients implemented in Java. The main disadvantage of CORBA however, is that being both language- and platform- independent, along with almost two decades of new features [4] while maintaining backwards compatibility have led to a large, complex system. This leads to an exhaustive amount of specifications, and steep learning curve. Also, the interoperability of CORBA becomes questionable as few vendors implement the entire functionality of the protocol.

The Internet Communications Engine, or ICE, was created by a small group of influential CORBA developers [9], and is both simpler and smaller than CORBA. Smaller, in this case, is a good thing, as few CORBA implementations have total coverage of the protocol. ICE also has a strong performance advantage [10].

2.3.6 Alternative ORBs

ORB:s are available from a large number of commercial and non-commercial organizations/vendors. Although OMG has defined standards in many areas for example security services, it is up to the vendor to provide an implementation. Usually, the implementations are simply a subset of the entire standard. Therefore, vendors differ in the functionality they provide. Often, open source ORBs are less complete, but using a commercial vendor is often an expensive alternative.

2.4 TMF814

TM Forum is one of the leading industry associations focused on IT for service providers in the communications, media, defense and cloud service markets [11]. Several major companies and the leading developers in the industry collaborate to develop standards. One of these standards is TMF814 Multi Technology Network Management (MTNM). The protocol is specifically created for telecommunication and supervision of equipment. It is comprehensive enough to satisfy all the members of TM Forum by supporting all the different manufacturers, and their respective equipment. The current version of this standard is TMF814 Multi Technology Network Management Solution Set 3.5. It contains of the following documents that
can be found by members at TM Forum:

- The Business Agreement - Document number TMF513
- The Information Agreement - Document number TMF608
- The Solution Set - Document number TMF814
- The Implementation Statement - Document number TMF814A

The Solution Set document contains the interface (IDL files) for the protocol. Therefore, an application that implements the MTNM Solution Set is said to support TMF814.

2.4.1 Structure

TMF814 is composed of multiple modules. Some of the bigger modules are managed by a manager. The entry-point and the module that ties everything together is called EmsSession. This module has the ability to provide the client with all the other managers, if they are properly implemented.

Figure 5: Managers and modules for TMF814.

Figure 5 illustrates the different modules that build up the TMF814 structure. EmsSessionFactory creates an EmsSession which is used to get access to the managers. The managers inside the blue area, e.g. EmsMgr, ManagedElementMgr, MultiLayerSubnetworkMgr and GuiCutThrough, are mandatory managers and should exist in all implementations. The other managers are optional. Modules that don’t have managers mostly contain data-structures and common data types. CORBA
modules (depicted in the orange area in Figure 5) are also found in the implementation code for TMF814.

2.4.1.1 EMS Manager
The EMS Manager represents the whole Element Management System. The subnetworks and multilayer-routing areas, including network elements (ManagedElements in TMF814) and the links between elements (TopologicalLinks), can be retrieved as lists by the client. This is the first step the client takes in order to find out which elements the server controls. It is also possible to request all of the alarms on the system (getAllEMSAndMEActiveAlarms and getAllEMSAndMEUnacknowledgeActiveAlarms), with some different functions to provide some basic filtering. A total of six different functions can be used for alarms on equipment that this manager controls. By using the correct function for the client’s specific need, the total amount of network traffic can be reduced. EMS Manager also includes functions for managing, creating and deleting Topological Links (as they don’t have a manager of their own). This also applies to Alarm Severity Assignment Profile (ASAP). The total amount of functions that this manager supports is 37, from which 5 are common functions that apply to all objects in this standard. A complete list of functions can be found in Appendix D.

2.4.1.2 MultiLayerSubnetwork Manager
MultiLayerSubnetwork is a way of grouping the functionality for subnetworks, which are called MultiLayerSubnetworks in TMF814. Each subnetwork may contain multiple elements. A basic functionality that this manager provides is providing the client with all the network elements within a specified subnetwork. All the Subnetworks and Subnetwork connections (SNC), are created managed and deleted from this manager. 19 functions are dedicated to management of SNCs in this manager. Routing and Calls are also handled here. The total amount of functions in this manager is 74, including the 5 common ones.

2.4.1.3 ManagedElement Manager
Each ManagedElement has several termination points, this is where actual connections start or end. Termination points exist in different forms depending on the equipment, but normally one ManagedElement consists of a set of Physical Termination Points (PTP) and a set of Floating Termination Points (FTP). PTPs and FTPs are physical ports (that you can plug a cable into) on the ManagedElement. Some equipment also has Contained Termination Points (CTP). CTPs are located on a specific layer rate and are always contained inside a PTP or FTP. Most of the functionality on this manager is devoted to terminations points, 25 of the total 40 functions. Other things that the manager handles are i.e. alarms on a specific element, cross-connections within the element and grouping of termination points (GTP).
2.4.1.4 EquipmentInventory Manager

Each Managed Element that is supervised by TMF814 can contain a different setup of equipment. The equipment inventory manager provides the necessities to model the different setups. Properties and placement are data that this manager handles for each equipment-card inside the specific element. It also manages the supported types of equipment in the different slots and holders, as well as the replacements and upgrades that are needed on a managedElement. A total of 21 functions constitutes this manager.

2.4.2 Entry-point when implementing

The entry-point for TMF814 is, as previously mentioned, the EmsSessionFactory. This is the object that the server registers in the Naming service, and thereby also the object that the client asks for when trying to connect to the server through the Naming service. When resolving object references in the Naming service TMF814 specifically requires the following NamingContext binding types:

- Class (context binding)
- Vendor (context binding)
- EmsInstance (context binding)
- Version (context binding)
- EmsSessionFactory (regular object binding)

Once the client has found the EmsSessionFactory, it invokes the operation getVersion() to determine the exact version of TMF814 that the counterpart supports and operates on. The set of functionality provided by the server differs between versions mainly in the sense that later versions extend the protocol with more functionality. When the version is known to the client, the client invokes getEmsSession(). This method creates a session between the server and the client. A user name and password can be supplied as arguments to this method to increase security.

When a valid session has been created, it is up to the client to decide which functions to use. If the client has support for and wants to use notifications to get events pushed to itself from the server, the getEventChannel function can be called from the session. This returns an event channel to the client, in which the server can send events. According to the TMF814 protocol a push-driven model should be used to minimize the stress on the server (which might be connected to multiple clients simultaneously.

Figure 6 shows the traffic between a client and a server when the session is being created. It is mainly composed of requests in coalition with the creation of the session. Prior to the traffic depicted, the client had requested the EMISSessionFactory from the NameService. Finally, the client can request whatever objects it wants. In
Figure 6: Example of how the client connects to the server. The blue thread depicts the notification channel.

In the case of figure 6 the client requests all subnetworks and their containing elements, because it wants to draw the hierarchy-tree at session connection.

From the session it is possible to request different managers. The function getSupportedManagers() provides the client with a list of which managers the specific
server actually has implemented and thereby supports. All managers are returned in a holder, Common_IHolder. The client can, after retrieving this holder, narrow the object to the specific manager that was requested. Figure 6 also shows the setup of the event channel. This part is threaded and can thereby occur at any time in aspect to getting subnetworks and elements.

Trying to use one function that is not yet implemented in the server will just give an unknown CORBA exception or if the server is implemented well, return the specific CORBA exception NOT_IMPLEMENTED. The first one is a general exception. This occurs if the server does not throw an exception on each unimplemented function. This exception can also mean that something unknown has happened during this transaction of information between server and client. The latter is a user exception which gives a hint to the client that the request was received, but the function is simply not implemented yet.
3 Method

When developers have little or no experience with one or more of the fundamental technologies within a project, ideas that are initially formulated might contain technical issues, conceptual flaws and/or implementation-related shortcomings. The time it takes to implement certain things can be difficult to estimate. Also, with vaguely specified requirements, the programmers must be able to have frequent communication with end-users of the product and have the ability to change requirements as the problem is more understood and solutions are revealed. A project is therefore naturally inclined to change throughout the development process.

Because the TMF814 is a fairly new competence-area for the GSDC OSS Integrations team, and because this project is conducted by a small and agile team of two, a linear process is not suitable. An iterative and incremental development model is generally good for agility [12].

The process would consist of three major iterations:

First iteration. This iteration includes coding the underpinnings and setting the architecture.

Feature-Complete Beta. The beta version includes all the major features required for user-testing.

Final product. This iteration includes final bug-fixes and adjustments based on analysis of test results.

In each iteration, a design phase is followed by an implementation phase followed by a testing phase. The first two iterations will have more emphasis on the design phase, while the test phase will simply be a quality review. The final iteration will use the design phase as an opportunity to receive feedback from the client in order to make priorities about what should be added or changed before the product is launched. The final test phase will include an acceptance test, but not unit tests because of the time limitations. The acceptance test will be a tool for future developers to grasp what is finished, what needs be fixed and what isn’t part of the final release.

Besides working incrementally, agility is achieved through the following methodology:

• Frequent releases

• Frequent communication with the client about needs

Maintainability is also important for this project, as the goal is to create a platform for future products. Therefore, these parts are emphasized:

• Transparency with the client about code and design choices
• Documentation
• Prioritizing simplicity and clarity in code

The project management process used in this project does not include pair-programming, but rather separating tasks, communicating, and making design decisions together. The benefit of pair-programming is a high quality code-review, but the benefit of separating tasks is speed, which was an unfortunate necessity when planning for a project with this scope and time-frame.
4 Design and Implementation

This section is separated into three individual parts: NBI, GUI, and the database, because of their modularity, and independence of each other. In figure 7 the different areas are color coded. The red part, NBI, could be replaced with another protocol-interface without having to change anything in the database or in the GUI. The same goes for the GUI and the database.

Figure 7: A simplified overview of the architecture of the simulator.
4.1 Data

This project involves simulating large amounts of network data, and naturally requires a database to organize the data and handle relations between the content. Our choice of database manager is MySQL, which, apart from taking the experience of the GSDC OSS Integrations team at Ericsson into consideration, has the advantage of high availability [13].

A script client in Ericsson’s portfolio extracts database information from ServiceOn servers and generates xml-files for each TMF function call. It does so by making requests for data over the TMF814 protocol. The data received is thus data that is accepted over the constraints of the protocol. By using the JavaScript Client, a parser can then extract the XML information into attributes that can be inserted into the internal database. Because access to the database is needed from the parser, from the northbound controller as well as from the GUI, it is good practice to wrap the JDBC function calls behind an interface.

Since the database interface contains all the functionality needed from both the GUI and NBI, a large amount of functions have to be implemented. Writing all the functions in a single class makes working with that class almost unfeasible. As the file grew it was decided to split the class into multiple classes without disrupting their interdependence. In this case, multiple inheritance would have been desirable, but Java lacks support for this. In order to be able to split the implementation into smaller files, chain inheritance is used. This is indeed an issue without a perfect solution. The downside to chain-inheritance used in this manner is that there might be an issue when adjusting dependent classes, and it abuses the nature of Java. Nonetheless, the same maintainability issues exist in large classes as well. Figure 8 shows how chain inheritance can look to solve this problem. The blue part is an interface, while the green parts are classes. The file at the bottom of the chain in Figure 8 implements the actual interface with all the functions. This file also inherits implemented functions from the class above, which in turn inherits from the class above itself, and works as a single access-point to all the functionality.

Figure 8: Chain inheritance.
In order to simulate TMF814 communication, and visualize it graphically data that complies with the TMF814 standards is needed. If the database does not define the correct constraints on the data, it is a risk that errors are not discovered until the client system cannot decode the TMF814 function. Debugging this way involves snooping the data traffic and does not always give an error message that can pinpoint the direct source of the problem. The primary constraints to be implemented are relational constraints.

Foreign keys prevent, for instance an alarm to be placed on a managedElement that does not exist, which would be difficult to represent graphically. When a managed Element is removed, all adjacent edges should be removed accordingly. Certain attributes are expected, according to the standards, to be, for example integers or values from a predefined list, although most attributes (such as owner) are simply strings. The database will not provide completely fail-safe type-checking as many of the attributes that pass through the protocol go under the generic type Any. These attributes are simply placed into the database with the VARCHAR type. Because the parser is extracting data that has been received over TMF814, it is safe to assume in these cases that the received attribute-types are correct. To summarize, the database will be strict on relational constraints but not type-constraints. Figure 9 shows a simplified schema over the database showing only tables and their relations.

Figure 9: The database architecture (only table names and their relations are depicted). See appendix A for complete schema.
4.2 GUI

The functionality required for the project requires data access code, north-bound communication code, and GUI code. In order to build a program that is as easy to maintain as possible, a good software architecture is needed. In this project, the data access code will be the center of the application, as it will go two ways: north-bound over TMF814, and straight to the GUI. It is therefore appropriate to create an interface between the database communication and the rest of the program.

GUI code can often be divided into data access code, logic code and presentation code. The model-view-controller software architecture is one of the most quoted architectures for graphical user interfaces and decouples the three types of code included in the GUI. The decoupling makes the code easier to follow, requires less copy-paste code, and makes small changes doable without having to change massive amounts of code due to all the different dependencies. There is simply a function call or an observable-observer relation between the blocks of code that one must keep in mind when making changes.

There are several different libraries available for graph representation in Java, and because these libraries are comprehensive and satisfactory for the need of this project, three of the more popular ones was reviewed; LGPL, JUNG and Annas. The choice to use JUNG (Java Universal Network/Graph Framework) was based on the ease of it’s use for the purpose of simply generating an attractive editable graph that contains node and edge information.

4.3 Northbound Communication

The decisions that must be made prior to the implementation of the north-bound communication are what ORB to use, and what Services to use. In order to avoid compatibility issues with TMF814 client programs within Ericsson’s portfolio, OpenORB is used. Because every vendor adds extensions to the base CORBA standards, interoperability problems can occur when IIOP is used to connect diverse ORBs. OpenORB includes its own services including Notify (a Notification service) and OpenORB Interoperable Naming Service (INS), which is a fully compliant implementation of the persistent Naming Service, INS specified by OMG.

The first step in the implementation process of the NBI is to compile the TMF814 IDL files and include them in the Naming Service. Compilation can be done for Java with Sun’s Java IDL (idlj). Then, we must initialize a CORBA connection. Before running a CORBA application, the ORB must be started and a port number must be given.
The sequence of actions performed in the connection consists of the following:

**Initialize the orb object.**

```java
String[] args = new String[] {"-ORBInitRef",
  "NameService=corbaloc::1.2@localhost:21234/NameService"};
orb = ORB.init(args, null);
```

**Reference to the root POA is retrieved and activated.**

```java
POA rootpoa = POAHelper.narrow(
  orb.resolve_initial_references("RootPOA"));
```

Because all Corba objects are generic, a function that looks like this are common after reference retrieval. The function narrow() is used to cast the object reference to the correct type. Activating the root POA causes associated POAs to start processing requests.

**Instantiate the servant object.**

Get the object reference associated with the servant. The object reference contains an address, the name of the POA that created an object reference and an object ID.

```java
EmsSessionFactory_IPOAImp emsSF =
  new EmsSessionFactory_IPOAImp();
org.omg.CORBA.Object ref =
  rootpoa.servant_to_reference(emsSF);
EmsSessionFactory_I emsSFRef =
  EmsSessionFactory_IHelper.narrow(ref);
```

**Obtain initial NamingContext and Register the servants with the Name Service.**

Below you will find the syntax that the simulator uses to add the EMSSessionFactory-object to the NamingService is shown. The model.getEmsName() function returns the name of the system that the simulator simulates.

```java
NamingContext ns = null, nc1 = null, nc2 = null;
ns = NamingContextHelper.narrow(
  orb.resolve_initial_references("NameService"));
nc1 = ns.bind_new_context(new NameComponent[]{
  new NameComponent("TMF_MTNM", "Class")});
nc2 = nc1.bind_new_context(new NameComponent[]{
  new NameComponent("Ericsson", "Vendor")});
```
nc1 = nc2.bind_new_context(new NameComponent[]{
    new NameComponent(
        model.getEmsName(), "EmsInstance")
});
nc2 = nc1.bind_new_context(new NameComponent[]{
    new NameComponent("3.5", "Version")
});
nc2.rebind(new NameComponent[] {
    new NameComponent(model.getEmsName(),
        "EmsSessionFactory_I"),emsSFRef
});

Wait for invocation.

The initial communication between the client and the server begins with the client requesting the _is_a() function. The function simply verifies that a CORBA object is an instance of a class that implements the correct interface. It then requests a list of the nameBindings in the Naming Context on which the operation is called with the function list(). Once all the name bindings are listed, it can iterate through them by requesting next_one(). On each name binding, the operation resolve() is used to return the reference bound to the specified name.

As expected (described in chapter 2.4.2), the client sent requests to resolve references to the following names (id : kind) in their respective order:

- Class (context binding)
- Vendor (context binding)
- EmsInstance (context binding)
- Version (context binding)
- EmsSessionFactory (regular object binding)

The client now has a reference to a TMF814 object. Because all the servant files are initially empty, it is up to the developers to implement the desired functionality.

Once the client has a reference to EmsSessionFactory it will call the following function which should return a reference to emsSession (by setting the parameter emsSessionInterface):

getEmsSession(String user, String password,
    NmsSession_I client,
    EmsSession_IHolder emsSessionInterface)

Now the client has a reference to emsSession which provides the all the session-related functionality, including:

getManager(String managerName,
    Common_IHolder managerInterface)
Which returns a reference to the given manager. As explained in section 2.4, there are several managers in TMF814 that hold functionality for different parts of the network. Now, which methods will be called, depends on how the client system is developed and how it is used. The client might for example want to retrieve all ManagedElements. The client would in that case request managedElementMgr and call the function:

```java
public void getAllManagedElements(int howMany,
    ManagedElementList_THolder meList,
    ManagedElementIterator_IHolder meIt)
    throws ProcessingFailureException {
```

As shown in the example function above, the function has no return value, but rather parameters that expect content. The client defines a maximum number of managedElements to retrieve by providing an int in the parameter howMany. meList is where the server will place the list of ManagedElements, meIt will contain the iterator that the client should use if they were to decide that they want to retrieve more.

The simulator would begin by retrieving all the managedElements from the database. Then, it would create a list of objects of the type ManagedElement_T (specified by TMF814) by converting each attribute to the correct type and instantiating the object. Because certain types have several layers of custom types, creating the correct type can require up to a few hundred lines of code.

Then a new ManagedElementIterator is created. The reference to the iterator is retrieved, which gives a handle to the object and the next_n can be called with howMany and meList as arguments. The code can be seen below:

```java
it = new ManagedElementIterator_IPOAImp(list, nbi, howMany);
Object iterRef = poa.servant_to_reference(it);
meIt.value =
    ManagedElementIterator_IHelper.narrow(iterRef);
meIt.value.next_n(howMany, meList);

nbi.managedElementIterators.add(meIt.value);
```

next_n will then, depending on the integer value in howMany, remove a certain amount of ManagedElements from it’s internal memory list and set: meList.value = list. Now that the values meIt and meList are set by the server, the client will receive these values.

---

1The client retrieves more managedElements by calling `next_n(int howMany, ManagedElementList_THolder meList)`
The structure of the TMF814 interface implementation might seem slightly tedious, but the structure of the implementation will look similar for every object. The only difference is the set of object types that builds up the element.
5 Results

The final product produced in this project is as planned a standalone OSS simulator. It can import foreign data from command-line as well as from the GUI (see figure 10). Figure 10 to 14 shows some of the views in the simulator.

Figure 10: This is how the GUI looks when importing data from XML-files.
Figure 11: An example on how the graph could look with seven managed elements and seven links between them.

Figure 12: The Real Time Alarm View, showing alarms of different severity types.
Figure 13: This is the Termination Point view, showing all the TPs for the selected managed element.

Figure 14: Showing the form for creating an alarm on the selected managed element. This view is also used to alter and update alarms.
5.1 Test Results

In the final iteration, we produced an external acceptance test document (see Appendix B). The document gives an overview of the functionality that works, that still needs work and that has not yet been implemented. Note that the scope of the functionality was altered as late as the final design phase. In the figure 15 below, you can see the proportions of finished, incomplete and unimplemented functionality in final release of EOS. If we recognize all three parts of our project as equally important, we have accomplished to implement 82 percent of the desired functionality.

Figure 15: Finished, incomplete and unimplemented functionality according to the Acceptance test for EOS in the third iteration. The pie charts are separated by sections of the architecture.

Figure 16 shows the correspondence of finished, incomplete and unimplemented functions in EMS Manager, ManagedElement Manager and MultilayerSubnetwork Manager. Implementing the protocol entirely would mean implementing all the functions in all the Managers.

Figure 16: Finished, incomplete and unimplemented functionality in three of the managers in TMF814
6 Discussion

6.1 Problems

In this section, some of the complications experienced during the design and implementation of EOS will be covered.

The first roadblock was discovering how to get a handle on the first object over CORBA. As described in chapter 2.3, an object is requested from the server to initialize the connection between the client and the server. The server should produce an IOR file for this object, but it is important that it is the right object. The first object that should be requested is specified in the protocol, but finding specifications for this in the extensive documentation for TMF814 was not completely effortless. After reading parts of the documentation, and capturing traffic between the GUI client and a real system it eventually became clear that the first object that the client will request from the server is EMSSessionFactory. The factory can then produce an EMSSessions between the server and the client which is needed for setting up a connection.

The next challenge was distributing the IOR information from the server to the client that wants to connect. The simplest way of doing this the first time a connection is being created is just to save the information to a file when the server starts, and then point the client to this specific file. This is a fast way of testing your connection in the beginning but of course this is not something that is sustainable for larger systems. The solution for solving this so called Bootstrapping problem, that is, how finding each other for the first time over a network is solved by NameService. This service acts almost as a digital phone-book. If the name and path to the name are known then the only thing left is to know where the NameService is located. The service can handle several CORBA objects, and can be updated by the servers to always have the newest IOR information stored. The location of the service is public information that can be retrieved by any client.

The NameService has as described before a lot in comment with an ordinary file-system. Unfortunately the service is not as advanced as a file-system, which makes it challenging. With the right commands it is possible to:

- search each NamingContext
- add/bind objects to current NamingContext
- add NamingContext to current NamingContext
- update object information
- remove/unbind object
- remove NamingContext

The slightly limited functionality, compared to a regular file-system, makes it
a little bit harder to work with. The fact that NamingContexts are just NamingContexts and not directories makes them a little bit different to work with. The documentation regarding the used service from OpenORB is not up to date at the time of writing and a lot of information is missing. Learning how to use this service thereby requires try and fail before the correct syntax is found so that everything complies with the TMF814 standard as described above.

A lesson that also was learned was that the protocol is strict about converting the TMF814 objects to the right types. TMF814 has many of its own types, and most of them are simple in nature, but a few contain several layers of custom data types. Also programming for CORBA requires that you put your programming language mentality behind, and think CORBA. Functions in TMF814 when compiled from the IDL-files do not have a return types, something that Java normally has if values are to be returned. The values are instead returned inside holders that are passed along as arguments to the function. It’s also important to remember that everything is focused around the ORB. For example, one could normally create an object of type Any with the following code:

```
Any a = new Any();
a.insert_Value(V v)
```

But CORBA requires the following:

```
any = nbi.getOrb().create_any();
V_THelper.insert(any, v);
```

When working with a system that is dependent on input from another system, the transition must be performed gracefully. Another fundamental issue experienced during the course of this project was to actually have input data. The data used was created by the previously mentioned Script Client that generated XML files. This data was however not complete until the end of the project, which made it hard to know exactly how the correct data could look and how it should be handled. Some of the scripts that the Script Client uses had to be rewritten to work properly, something that was outside the original scope but was necessary to be able to retrieve real correct input data.

To be able to display all the data from the database in an efficient and user-friendly way a lot of the models for standard objects had to be rewritten. Standard cells in JTable do not change color depending on the value of the perceived severity in the alarm table case. A custom model for displaying data had to be written. Even third party libraries had to be rewritten to some extent to be able to get the best result.

Since the architecture requires that northbound TMF814 should be a separate part, the program loads all modules that implements NBI.java at startup, if no interface is found then only the GUI- and database parts will be available to the
user. This proves that the different parts actually are modules. Other protocols that looks almost like TMF814 could also be used as long as they implement the given interface. The northbound interface for TMF814 is managed and developed as a different project, which is loaded in to the simulator through dynamic loading of classes in Java.

One other problem that occurred during this project was finding the exact way that alarms should be created. Alarms are specified with the CORBA-specific type Structured Event. The structure is defined and documented, but many of the fields are built up with the type called Any. This type is basically a container for any other type or object. This makes Java’s type checking unable to verify that everything inside the Any-object correct according to the CORBA structured event type. If an alarm is created with faulty data in one of the Any-objects, errors will occur on the client side, but the server will not be notified. Because the clients used for testing were not implement during this project, it made troubleshooting very hard and time consuming. It was not straightforward whether the clients implementation was faulty, the version was different or our data was built the wrong way.

The software engineering method used was generally very good, but 4-5 iterations would have been better for the scope of the project. This mostly because of the steep learning curve in the beginning, which resulted in much of the massive functionality being implemented in the final iteration. This led to a very relevant test phase being left to be handle by those who choose to keep developing the product.

### 6.2 Frameworks and Library choices

The choice of programming language for this project is, as always, questionable. Being a standalone product, the only initial restrictions on the language were effective communication means to large sets of data and CORBA-support. Ease of access to the runtime environment as well as client competence (potential team of future developers) also had an important role in the choice of language. With multiple languages supporting the first two restrictions, a choice was made based on the simplicity and competence factors. Nonetheless, Java has some limitations that have effected our project. E.g. Java lacks multiple inheritance. When a single class became too large to be manageable, and functions within the class were inter-dependent, the only options are to simulate multiple inheritance. This is done by allowing an interface to extend multiple interfaces, which in turn are implemented. This would cause the library to be filled with twice as many files. A choice that might seem best for the object-oriented fundamentalist, as well as for the architect, but inconvenient for the developer.

In the industry, NoSQL databases are becoming more relevant alternatives to classic relational databases because of their speed, agility and the fact that they support a distributed architecture [14]. While speed is important in EOS (retrieving thousands of contained terminations points can cause delay), neither agility or
distribution are arguments that are important enough in this project to choose a non-relational storage type. NoSQL, requires that much of the logic that is handled by databases is programmed on a higher level. Also, because this project requires data from client SQL databases and is restricted by the standards of TMF814, the data received is structured in a way that is appropriately placed in a relational database. Because of the simple hierarchical nature of the data in TMF814, the requirements needed for the project were supported in both PostgreSQL and MySQL, and performance differences between the two are small enough that the choice of MySQL in this project had no obvious negative impact.

Although using CORBA was not a active choice, it is indirectly a choice made by TM Forum due to the fact that TMF814 is based upon CORBA, and it is interesting to see how this architecture has effected the progress of the project. As mentioned in chapter 2.3, CORBA has a steep learning curve because of the massive amount of functionality to filter through before grasping the relevant information. Naturally, documentation of this size rarely are up to date on every aspect, which was the case for CORBA at the time of writing.

In chapter 2.3.1, we mentioned that the object oriented approach is good for protocols where the client and server know exactly what to expect of each other (strictly defined in the protocol), so interfaces can simplify the development. Our experience is that this simplification refers to development in the long-term. Setting up the CORBA connection and getting a handle on the first object is not a simple process. TMF814, is not a small protocol, however the steep learning curve, and the time it takes to initialize a connection would make it pointless to make a small protocol based on CORBA. The advantage of using a CORBA based protocol is that now when the setup is complete, the security and connection is encapsulated from the developer and all we have to think about is implementing the methods. It is as if the objects handled are local.

Another negative aspect experienced was the unnatural mapping to Java. Compiling the IDL files to java classes worked well, but the code was not what the Java developer would naturally expect. One example was that all the functions had the return type void. Instead of returning an object, the parameters of the function contained values that the server had to set.

As mentioned previously, IIOP interoperability has not always been failsafe. Basic ORB functions will work cross vendor, but because of extensions to the standards, a problem could arise in the future of the product due to this. Also, if the TMF814 protocol changes, which it is prone to do, both the server and the client must recompile their IDL files. Typically the server will crash if they have different versions of the protocol.

The experience with TMF814 is that the specifications are well-written and fairly well-structured, but that guides and tutorials are missing. Also, the type specifications cover the entire protocol, which forces the new developer to filter lots of information before finding certain fundamental type descriptions and action de-
scriptions. There is a forum, but most of the active members discussed very specific questions, and there was no thread for beginners. Type and action specifications is really what defines the protocol, but the types defined can cause problems in their mapping to some languages, in this case Java. There are no information about the protocol available if you are not part of the forum, TM Forum.

The CORBA standard and TMF814 both have something in common. They are complex, and comprehensive. In the case of CORBA, the promise of seamless interoperability is threatened by incomplete ORB implementations. In the case of TMF814, seamless interoperability is threatened by incomplete protocol implementations. This would not be an issue if the server and client were implemented by the same developers, but that prerequisite and our experience with the time it takes to implement a subset leaves us questioning the quality of the standard.

Legal limitations should also be taken into account. In April 2009, Sun and Oracle announced an agreement regarding Oracle’s acquisition of Sun. Sun has an early history of lawsuits, a privilege that the new owner knows how to take advantage of. In August 2010, Oracle announced a patent infringement suit against Google for 7 violations of Oracle’s Java-related intellectual property and a copyright infringement. Although the Java license is free and under a GPL license (apart from the Enterprise and ME version), the license does not allow sub-setting or super-setting (adding anything on top of the specification or implementing only parts of the specification). In Google’s case, it was mainly regarded their virtual machine for Android called Dalvik, which Oracle claims contains customized JVM code. How Oracle chooses to interpret deviation from the specification is not completely clear. Does it include customizing libraries? Although this project does not deviate from the license conditions, and will not be a consumer product, it is important to see that there is a considerable degree of legal uncertainty that should be taken into account.

Java is not the only technology in which a question of reliability arises. By buying Sun, Oracle also acquired MySQL, an important competitor to their main product - Oracle Database. With two competing technologies under the same company the competition becomes threatened. Although Oracle has promised to continue their support for MySQL, chances are that the open-source community contributing code to the project might diminish. Therefore, the future remains uncertain. The licensing policy of MySQL states that upon developing and distributing open source applications under the GPL or OSI-Approved License, no fees must be paid. [13] Again, developing a product for internal use is not affected by the license, but it is important to bear in mind for future strategic decisions.

6.3 Design Choices

Once the project had gone on for a while, it became clear that some of the files would become very large, making them almost impossible to work with. As men-
tioned earlier in chapter 4.1, Java does not support multiple inheritance. Splitting up the implementation of the database interface for MySQL was really necessary to make it workable. One could discuss if this is the correct way of solving the problem with respect to the Java best practices. In our case, not splitting up the file would have resulted in a large file that grows linearly with the scope of the project. Working with large files makes development inefficient. It’s hard to find relevant sections of code, and it can easily become unstructured.

When it comes to the GUI design and the general look and feel, it was decided that the simulator should resemble the system that was simulated. This makes it easier to use for someone who is familiar with ServiceOn. A tree to show structure of subnetworks and elements, but with the big difference that TPs where moved to a separate main-window. A managed element could have many thousands of TPs, which would make it hard to use in an ordinary tree. Our solution to this was to show TPs in a table instead, which makes it easier to use and work with. The table handles large amount of data in an more efficient and user-friendly way. Even the alarms are shown in a table in the main area. This area is the biggest area, making it possible to show the most data here. This table was equipped with filtering capabilities much like in ServiceOn to making it easier to find the specific alarms. Regular expressions could be used for fast filtering on all the data in the table.

To be able to meet the initial requirements regarding modularity and abstraction interfaces were developed between the clear parts of the simulator. One interface towards the database and one towards northbound communication. This means that without changing anything in the program itself the database could be swapped for another database, as long as the new one implements the interface. This is of course true even for northbound communication. The simulator could handle many NBIs at the same time, all because of the interface between the simulator and the actual northbound communication. The simulator is even so modular that it handles adding and removing NBIs without having to compile anything, as long as all NBIs specified follows the interface.

6.4 Concept and Client Value

Initially, it was unclear as to which extent of value the product would generate. Through frequent communication with Ericsson, this project has been able to take shape into a useful framework. The simulator is a product which Ericsson will be able to use internally for testing and perhaps externally. At times, only parts of the product might be useful, for instance reading client data locally and displaying it graphically. It can also be used simply to create objects and send it over TMF814. Because the GSDC OSS Integration is composed of people with diverse developing experience, certain members will be ready to use the product right away. However, those who do not have the time or experience to customize our implementation programmatically would appreciate a launch where more testing has been performed.
to minimize the risk for unwanted surprises. Refining the current product with the current available functionality for sales purposes would require 80-240 hours of development depending on the requirements of the client. Because the learning curve for development over CORBA is steep and bends, a loss of competence is difficult to replace with documents and would therefore prolong the process.

6.5 Future potential

There is a massive amount of functionality that could be added to the final product. The first step in continuing with the development of the simulator would be to go through one more iteration to polish the current functionality. When this is done, the developer in question might want to spend time on making the parser more intelligent. The optimal parser would take in everything it could, and if an attribute does not comply with the standard, it could give the user a choice as to whether or not it should be imported. Error descriptions could be better. Because we never know what kind of data the client system has, it is especially critical that the parser is tested with multiple systems. When it comes to the graphical user interface the next step to take would be to perform usability tests and undergo another iteration. The Northbound interface code can be enhanced with more implemented functions. However, as the results imply, implementing the entire TMF814 protocol would take a long time. Also, extending the NBI with functionality means that extensions must be made in the GUI and DBI accordingly (unless the intentions are to use the NBI implementation independently).
7 Conclusion

One of the main purposes of this thesis was to develop a deeper understanding of TMF814 and the surrounding protocols. We have gained experience throughout the product cycle, and a summary would be that we have worked with a protocol which requires lengthy development time in relation to the functionality achieved. Both CORBA and TMF814 require a combination of theoretical experience, patience and perseverance, but in the long run, a comprehensive middleware will be able to carry the information necessary for Network Management.

The other main purpose with this project was to build a stable architecture. Being able to attach and detach modules was something that we based the architecture upon, and we can now see on the final product that we indeed have a modular structure. Along with Modularity, we have built the implementation to be modified easily without affecting other parts of the system, and thus achieved encapsulation of the more complex logic. Although we have not had any problems with the performance or the usability of our simulator, we have realized the importance of testing. We have only run our implementation on three different laptops, with three different sets of client data, towards three different clients. There is still much to do before we can say anything about performance.

Usability testing always has a strong positive effect on the value of the product, so this is also something that should be performed in the near future. Our main efforts for maintainability involved clean code, open communication, comments, javadoc and a manual. Through frequent communication with Ericsson we have adjusted the product according to evolving needs and been able to produce a product that can be and already is valuable to them in multiple ways. It is also a platform that can be used for alternative purposes. We succeeded in importing and visualizing real network data, which alone is useful when preparing for an integration. Having a lightweight TMF814 implementation that the GSDC OSS Integrations can customize and test upon is also an excellent way to build knowledge within the group and prepare for a real solution.
References


Appendices

A - Database Schema
B - Test Results
C - Requirement Specification
D - Java-Doc
E - User manual
Appendix A – Database Schema
Appendix B – Test Result
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>P</th>
<th>F</th>
<th>N/A</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 DATA</td>
<td></td>
<td></td>
<td></td>
<td>The parser has been tested on data created in the simulator, extracted through the Script Client and then parsed through the parser. Data from real systems has been tested as far as we have got our hands on such data.</td>
<td></td>
</tr>
<tr>
<td>1.1 Parser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import EMS information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import Subnetworks from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import ManagedElement information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td>The script produces multiple files with MEs, one for each subnet.</td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import Topological Link information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import PTP information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import CTP information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td>If you use the custom javascript, it will produce several CTP files</td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import All EMS and ME Alarm information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td>The parsing works (But the Script Client can’t serialize structured events at the moment, but that is outside of the scope for this project.)</td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import Cross Connection information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Parse</td>
<td>Import SNC information from xml-files produced by the script-client through the command-line interface.</td>
<td></td>
<td></td>
<td></td>
<td>Not implemented in parser yet.</td>
</tr>
<tr>
<td></td>
<td>Data import can be performed from the GUI.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Warnings in GUI-import when a file is corrupt, or cannot be read.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Warnings in GUI-import when data does not comply with TMF814 standards, for example, an alarm on a managedElement that does not exist.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### GUI Display

The GUI displays correctly how many objects were imported into the database.

An object is not imported if a regular attribute is wrong, but if one of the attributes that has several values (such as additional Info) fails, the object is STILL imported, and just skips the additionalInfo and sends a warning, but on the GUI it will say for example 4 out of 5 objects were parsed.

Custom javascript for scriptclient that generates correct xml file for corresponding info.

The javascript file for managedElement does not handle the data optimally, this is because it requests all subnetworks first, followed by all managedElements under that parser. The xml data produced by this is not parsed easily, so an quick solution was to split the data into files for all managedElements under a single subnetwork.

### Panel

<table>
<thead>
<tr>
<th>File &gt; new Project: should empty the database</th>
<th>A popup should warn the user that the database will be emptied!</th>
</tr>
</thead>
<tbody>
<tr>
<td>File &gt; save Configuration: should save debug level, alaramorder to Config.java</td>
<td>A popup could appear afterwards saying &quot;Config.java has been saved&quot;</td>
</tr>
<tr>
<td>File &gt; reset configuration: should reset Config.java</td>
<td>A popup could appear afterwards saying &quot;Config.java has been reset&quot;</td>
</tr>
<tr>
<td>File &gt; realign: should sync the GUI with the database content</td>
<td>A popup could appear afterwards saying &quot;EOS has synchronized with the database&quot;</td>
</tr>
<tr>
<td>MainView &gt; RTAM: will display RTAMView</td>
<td></td>
</tr>
<tr>
<td>MainView &gt; TP: will display all TPView</td>
<td></td>
</tr>
<tr>
<td>MainView &gt; Map: will display MapView</td>
<td></td>
</tr>
<tr>
<td>Northbound Interface shows up if there is a zip file in the NBI directory</td>
<td></td>
</tr>
<tr>
<td>NBI &gt; connect</td>
<td>Not able to unregister the EMS from NameService</td>
</tr>
<tr>
<td>NBI &gt; disconnect</td>
<td>This works, but we have only one NBI to test on, it should work for multiple NBIs</td>
</tr>
<tr>
<td>NBI &gt; connect all</td>
<td></td>
</tr>
<tr>
<td>Services &gt; Start: should restart Notification and Naming Services</td>
<td>Only tested for the TMF814 interface</td>
</tr>
<tr>
<td>Services &gt; Restart: Notification and Naming services should be restarted</td>
<td>TODO in MainController.java row 196</td>
</tr>
<tr>
<td>Services &gt; Stop: Notification and Naming services should be stopped</td>
<td>TODO in mainController.java row 192</td>
</tr>
<tr>
<td>Help &gt; About page</td>
<td>Implemented, but should be filled with meaningfull text</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Help &gt; JavaDoc (for EOS)</td>
<td>TODO in MainController.java row 235, also, change name from OSSSim to EOS</td>
</tr>
<tr>
<td>Help &gt; JavaDoc (for TMF814)</td>
<td>TODO in MainController.java row 244</td>
</tr>
<tr>
<td>Help &gt; visit homepage and Help &gt; visit TMForum homepage</td>
<td>—</td>
</tr>
</tbody>
</table>

### 2.2 TreeView

<p>| Ems name shows up in the treeView | — |
| Subnetworks show up in the treeView | — |
| ManagedElements show up in the treeView | — |
| ManagedElements with a heartbeat are green, otherwise red | — |
| right-click on EMS &gt; EMS info: lets you view and configure EMS info | — |
| right-click on EMS &gt; Create MultiLayerSubnetwork | — |
| right-click on EMS &gt; Create Alarm/TCA | — |
| right-click on a MultiLayerSubnetwork &gt; EMS info | — |
| right-click on an MultiLayerSubnetwork &gt; MultiLayerSubnetwork info | — |
| right-click on a MultiLayerSubnetwork &gt; Create Alarm/TCA | — |
| right-click on a MultiLayerSubnetwork &gt; Delete MultiLayerSubnetwork | — |
| right-click on a ManagedElement &gt; EMS info | — |
| right-click on a ManagedElement &gt; MultiLayerSubnetwork info | — |
| right-click on a ManagedElement &gt; ManagedElement info | — |
| changing the name of a ManagedElement in the tree updates the map | — |
| right-click on a ManagedElement &gt; Create Alarm/TCA | — |
| right-click on a ManagedElement &gt; Create PTP/FTP | — |
| right-click on a ManagedElement &gt; HeartBeat &gt; Disable/Enable (should change the connectionState) | — |</p>
<table>
<thead>
<tr>
<th>2.3 RTAM</th>
<th>RTAM view lists all active alarms on EMS, Subnet and ME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>double-click on an alarm lets you view and configure alarm info</td>
</tr>
<tr>
<td></td>
<td>right-click on an alarm &gt; Alarm Info: lets you view and configure alarm info</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; Acknowledge and Unacknowledge alams</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; Delete Selected Alarms</td>
</tr>
<tr>
<td></td>
<td>RTAM view columns in relevant order: source of alarm, severity, probable cause..</td>
</tr>
<tr>
<td></td>
<td>column names in RTAM view are understandable and relevant</td>
</tr>
<tr>
<td></td>
<td>perceivedSeverity generates different colored rows</td>
</tr>
<tr>
<td></td>
<td>Filtering on NT_Alarm or NT_TCA</td>
</tr>
<tr>
<td></td>
<td>Filter with either AND or OR between</td>
</tr>
<tr>
<td></td>
<td>Filter on ObjectType</td>
</tr>
<tr>
<td></td>
<td>Filter on AcknowledgeIndication</td>
</tr>
<tr>
<td></td>
<td>Filter on ServiceAffecting</td>
</tr>
<tr>
<td></td>
<td>Filter on Severity</td>
</tr>
<tr>
<td></td>
<td>Attributes with multiple values should display content by holding the mouse over the field</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.4 TPView</th>
<th>When an ME is marked, all contained PTP/FTPs are listed in TP view</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For each PTP/FTP, all potential CTP’s are listed in TP view</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; TP Info</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; ManagedElement Info</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; EMS Info</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; Create CTP</td>
</tr>
<tr>
<td></td>
<td>right-click &gt; Create Alarm</td>
</tr>
</tbody>
</table>

<p>| 2.5 MapView | Scrolling zooms in and out |</p>
<table>
<thead>
<tr>
<th>Action</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>right-click on ManagedElement &gt; ManagedElement Info</td>
<td></td>
</tr>
<tr>
<td>right-click on ManagedElement &gt; Delete</td>
<td></td>
</tr>
<tr>
<td>right-click on ManagedElement &gt; Create Alarm</td>
<td></td>
</tr>
<tr>
<td>right-click on ManagedElement &gt; Connection: should change the connectionState, which should change the icon color in the tree and map</td>
<td></td>
</tr>
<tr>
<td>right-click on TopologicalLink &gt; TopologicalLink Info</td>
<td></td>
</tr>
<tr>
<td>right-click on TopologicalLink &gt; Delete: should effect both the map and tree</td>
<td></td>
</tr>
<tr>
<td>configuring the name of an ME in the graph changes the name in the tree view</td>
<td></td>
</tr>
</tbody>
</table>

Transforming mode allows you to move the background by dragging
Picking mode allows you to move the managedElements and mark multiple ME's
Ctrl + left click on an ME centers it on the screen (in Picking mode)
Clicking on a Topological Link highlights it (in picking mode)
Editing mode allows you to left-click on the field to create a new managedElement
Dragging and dropping between two managedElements creates a new topological Link
Dragging and dropping on a single managedElement creates a topologicalLink that loops

2.6 Panels: common

Every time we chose "new _ " in a menu, a respective panel will popup where you can provide information
Editing LayerRates
Editing Additional Info

Adding a short number that is not predefined is not written to the database
### Adding wrong attributes should give a warning or error message

- **Cancel**: do nothing and close window
- **Restore**: restore to original values, keep window open
- **Save**: Insert data in the database
- **Error message should be sent if trying to save without primary key**

Data truncation should be displayed when data is too long for column

In debug window

---

### 2.7 Status

In the lower left corner of the GUI three icons represent the status of the following by changing icon color:

- **the database connection**
- **if Naming and Notification Services are running**
- **connection to the northbound interface**

---

### 3.0 NBI

#### 3.1 GUI Client

| Tree of EMS, MLSN, ME, PTP, CTP shows up in GUI | [ ] |
| Simulator can be found by login in GUI Client | [ ] |

#### EMS

| Open EMS gives the same info as shown in simulator | [ ] |
| Launch Network Data List on EMS shows a list of all ME in simulator | [ ] |
| Launching Monitor shows all alarms under the EMS in simulator | [ ] |
| Launching MultilayerSubnetwork on EMS shows all MLSN from simulator | [ ] |
| Launching Link shows all the Topological links under the EMS | [ ] |
| Launching SNC shows all SNCs under the EMS | [ ] |

| EMS Capabilities shows all the functions supported by the simulator | [ ] |
| “Details” show all the details as when connected to the simulator for the given MLSN | [ ] |

Alarms not working, problem with structured events

Not implemented in NBI

Not Implemented in GUI, model or NBI

Needs to be checked, but it shows what's programmed in the different mgrs.
<table>
<thead>
<tr>
<th>Get Object</th>
<th>get the info about the MLSN as shown in the simulator at the moment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lunch Network Data List</td>
<td>shows a list of all the Mes under this MLSN in simulator</td>
</tr>
<tr>
<td>ME Details</td>
<td>shows all the details as when connected to the simulator for the given ME</td>
</tr>
<tr>
<td>Get Object</td>
<td>get the info about the ME as shown in the simulator at the moment</td>
</tr>
<tr>
<td>ManagedElement Info</td>
<td>shows a list with everything related to the ME</td>
</tr>
<tr>
<td>PTP/FTP Details</td>
<td>shows all the details as when connected to the simulator for the given TP</td>
</tr>
<tr>
<td>CTP Details</td>
<td>shows all the details as when connected to the simulator for the given CTP</td>
</tr>
<tr>
<td>Get Object</td>
<td>get the info about the CTP as shown in the simulator at the moment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1.2 Script Client</th>
<th>Script-results compared to data in Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>001-getAllEMSAndMEActiveAlarms</td>
<td>No alarms work, problem with Structured Events</td>
</tr>
<tr>
<td>003-getAllTopLevelSubnetworkNames</td>
<td></td>
</tr>
<tr>
<td>004-getAllTopLevelSubnetworks</td>
<td></td>
</tr>
<tr>
<td>008-getEMS</td>
<td></td>
</tr>
<tr>
<td>016-getAllCrossConnections</td>
<td>In DB but not implemented in NBI</td>
</tr>
<tr>
<td>017-getAllManagedElementNames</td>
<td></td>
</tr>
<tr>
<td>018-getAllManagedElements</td>
<td></td>
</tr>
<tr>
<td>020-getAllPTPs</td>
<td></td>
</tr>
<tr>
<td>022-getContainedCurrentTPs</td>
<td></td>
</tr>
<tr>
<td>024-getContainedInUseTPs</td>
<td>CTP inside a SNC should be marked as InUse, we lack this feature</td>
</tr>
<tr>
<td>026-getContainedPotentialTPs</td>
<td></td>
</tr>
<tr>
<td>137-getAllManagedElements</td>
<td></td>
</tr>
<tr>
<td>138-getAllManagedElementNames</td>
<td></td>
</tr>
<tr>
<td>143-getAllTopologicalLinks</td>
<td>Not implemented in NBI</td>
</tr>
<tr>
<td>148-getSNCs</td>
<td>Not implemented in NBI</td>
</tr>
</tbody>
</table>
Instead of including all the functions in the different managers and mark all the unimplemented functions (which are many), we will list the implemented and commenced managers below

**EMSMgr:**

Completed:
- acknowledgeAlarms
- getAllEMSAndMEActiveAlarms
- getAllEMSAndMEUnacknowledgedActiveAlarms
- getAllEMSSystemActiveAlarms
- getAllEMSSystemUnacknowledgedActiveAlarms
- getAllTopLevelSubnetworkNames
- getAllTopLevelSubnetworks
- getEMS
- getCapabilities

Commenced:
- setAdditionalInfo
- setNativeEMSName
- setOwner
- setUserLabel

**ManagedElementMgr:** - (obs tp functions don’t allow filtering on layerRate yet)

Completed:
- getAllActiveAlarms
- getAllFTPNames
- getAllFTPs
- getAllManagedElementNames
- getAllManagedElements
- getAllPTPNames
- getAllPTPsWithoutFTPs
- getAllPTPs
- getAllPTPsWithoutFTPs
- getContainedCurrentTPNames
getContainedCurrentTPs
getContainedInUseTPs
getContainedPotentialTPNames
getContainedPotentialTPs
getManagedElement
getTP

Commenced:
setAdditionalInfo
setNativeEMSName
setOwner
setUserLabel

MultiLayerSubnetworkMgr:

Completed:
getAllManagedElementNames
getAllManagedElements
getMultiLayerSubnetwork

Commenced:
setAdditionalInfo
setNativeEMSName
setOwner
setUserLabel
Appendix C – Requirement Specification
TMF814 Simulator Software Requirements Specification

TMF814 Simulator
Mikael Riedel, Louisa Luciani
Ericsson Lindholmen

E-mail Louisa: louisa.luciani@ericsson.com
E-mail Mikael: mikael.riedel@ericsson.com
Contents
1 Introduction......................................................................................... 6
  1.1 Purpose....................................................................................... 6
  1.2 Document Conventions............................................................. 6
  1.3 Intended Audience and Reading Suggestions............................ 6
  1.4 Stakeholders.............................................................................. 6
  1.5 Definitions and Abbreviations.................................................. 7
2 Overall Description ........................................................................... 8
  2.1 Product Perspective................................................................. 8
  2.2 Operating Environment............................................................ 8
  2.3 Design and Implementation Constraints................................. 8
  2.4 Documentation........................................................................... 8
3 Overview of the system..................................................................... 10
  3.1 System components................................................................. 10
  3.2 System dependencies.................................................................. 10
4 Functional requirements.................................................................... 11
  4.1 Model......................................................................................... 11
    4.1.1 Database storage............................................................... 11
    4.1.2 Save.................................................................................... 11
    4.1.3 Load.................................................................................... 11
    4.1.4 New project.......................................................................... 11
    4.2 Heart beat............................................................................... 12
      4.2.1 Heart beat behavior.......................................................... 12
      4.2.2 Enable heart beat.............................................................. 12
      4.2.3 Disable heart beat.............................................................. 12
    4.3 Notification............................................................................. 12
    4.4 Active alarm list...................................................................... 12
5 TMF 814.......................................................................................... 13
  5.1 Connect to GUI Client................................................................. 13
  5.2 EMS manager............................................................................ 13
    5.2.1 Function: getAllEMSAndMEActiveAlarms.......................... 13
    5.2.2 Function: getAllEMSSystemActiveAlarms............................ 13
    5.2.3 Function: getAllTopLevelSubnetworks................................. 13
    5.2.4 Function: getAllTopLevelSubnetworkNames.......................... 14
    5.2.5 Function: getAllTopLevelTopologicalLinks........................... 14
    5.2.6 Function: getAllTopLevelTopologicalLinkNames.................... 14
    5.2.7 Function: getTopLevelTopologicalLink.................................. 14
    5.2.8 Function: getEMS.............................................................. 14
    5.2.9 Function: acknowledgeAlarms............................................ 15
    5.2.10 Function: createTopologicalLink....................................... 15
    5.2.11 Function: deleteTopologicalLink........................................ 15
    5.2.12 Function: getAllEMSAndMEUnacknowledgedActiveAlarms.... 15
    5.2.13 Function: getAllEMSSystemUnacknowledgedActiveAlarms..... 15
    5.2.14 Function: unacknowledgeAlarms........................................ 16
  5.3 ManagedElement manager.......................................................... 16
    5.3.1 Function: getManagedElement.......................................... 16
    5.3.2 Function: getAllManagedElements..................................... 16
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.3</td>
<td>Function: getAllManagedElementNames</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Function: getAllActiveAlarms</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Function: getAllUnacknowledgedActiveAlarms</td>
</tr>
<tr>
<td>5.3.6</td>
<td>Function: getAllIPTPs</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Function: getAllIPTPNames</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Function: getAllFTPs</td>
</tr>
<tr>
<td>5.3.9</td>
<td>Function: getAllIFTPNames</td>
</tr>
<tr>
<td>5.3.10</td>
<td>Function: getAllIPTPsWithoutFTPs</td>
</tr>
<tr>
<td>5.3.11</td>
<td>Function: getAllIPTPNamesWithoutFTPs</td>
</tr>
<tr>
<td>5.3.12</td>
<td>Function: getContainedCurrentTPs</td>
</tr>
<tr>
<td>5.3.13</td>
<td>Function: getContainedCurrentTPNames</td>
</tr>
<tr>
<td>5.3.14</td>
<td>Function: getContainedInUseTPs</td>
</tr>
<tr>
<td>5.3.15</td>
<td>Function: getContainedInUseTPNames</td>
</tr>
<tr>
<td>5.3.16</td>
<td>Function: getContainedPotentialTPs</td>
</tr>
<tr>
<td>5.3.17</td>
<td>Function: getContainedPotentialTPNames</td>
</tr>
<tr>
<td>5.3.18</td>
<td>Function: getContainingTPs</td>
</tr>
<tr>
<td>5.3.19</td>
<td>Function: getContainingTPNames</td>
</tr>
<tr>
<td>5.3.20</td>
<td>Function: getContainingSubnetworkNames</td>
</tr>
<tr>
<td>5.3.21</td>
<td>Function: getTP</td>
</tr>
<tr>
<td>5.3.22</td>
<td>Function: setTPData</td>
</tr>
<tr>
<td>5.3.23</td>
<td>Function: createGTP</td>
</tr>
<tr>
<td>5.3.24</td>
<td>Function: deleteGTP</td>
</tr>
<tr>
<td>5.3.25</td>
<td>Function: getGTP</td>
</tr>
<tr>
<td>5.3.26</td>
<td>Function: getAllGTP</td>
</tr>
<tr>
<td>5.3.27</td>
<td>Function: getAllGTPNames</td>
</tr>
<tr>
<td>5.3.28</td>
<td>Function: getContainingGTP</td>
</tr>
<tr>
<td>5.3.29</td>
<td>Function: modifyGTP</td>
</tr>
<tr>
<td>5.3.30</td>
<td>Function: getPotentialFixedCCs</td>
</tr>
<tr>
<td>5.3.31</td>
<td>Function: getAllCrossConnections</td>
</tr>
<tr>
<td>5.3.32</td>
<td>Function: getAllFixedCrossConnections</td>
</tr>
<tr>
<td>5.3.33</td>
<td>Function: setGtpAlarmReportingOn</td>
</tr>
<tr>
<td>5.3.34</td>
<td>Function: setGtpAlarmReportingOff</td>
</tr>
<tr>
<td>5.3.35</td>
<td>Function: verifyTDMAssignment</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Function: getAllEquipment</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Function: getAllEquipmentNames</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Function: getAllSupportedPTPs</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Function: getAllSupportedPTPNames</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Function: getAllSupportingEquipment</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Function: getAllSupportingEquipmentNames</td>
</tr>
<tr>
<td>5.4.7</td>
<td>Function: getContainedEquipment</td>
</tr>
<tr>
<td>5.4.8</td>
<td>Function: getEquipment</td>
</tr>
<tr>
<td>5.4.9</td>
<td>Function: provisionEquipment</td>
</tr>
<tr>
<td>5.4.10</td>
<td>Function: setAlarmReportingOff</td>
</tr>
<tr>
<td>5.4.11</td>
<td>Function: setAlarmReportingOn</td>
</tr>
<tr>
<td>5.4.12</td>
<td>Function: unprovisionEquipment</td>
</tr>
<tr>
<td>5.4.13</td>
<td>Function: getSupportedEquipment</td>
</tr>
<tr>
<td>5.4.14</td>
<td>Function: getSupportedEquipmentNames</td>
</tr>
<tr>
<td>5.4.15</td>
<td>Function: getSupportingEquipment</td>
</tr>
<tr>
<td>5.4.16</td>
<td>Function: getSupportingEquipmentNames</td>
</tr>
</tbody>
</table>
6 Nonfunctional requirements .............................................. 28
   6.1 ServiceOn resemblance ........................................... 28
   6.2 Nonfunctional requirements for TMF 814 .................. 28
7 GUI ........................................................................... 28
   7.1 Exit ................................................................. 28
   7.2 Switching view ................................................. 28
   7.3 Resizing ......................................................... 28
   7.4 Menu .............................................................. 29
   7.5 Map ................................................................. 29
   7.5.1 Adding NE ............................................... 29
   7.5.2 Moving NE ............................................... 29
   7.5.3 Removing NE .............................................. 29
   7.5.4 Adding TL ............................................... 29
   7.5.5 Removing TL .............................................. 29
   7.5.6 More info on NE ......................................... 30
   7.5.7 Showing alarms ......................................... 30
   7.5.8 Heart beat failure ........................................ 30
   7.6 RTM .............................................................. 30
   7.6.1 Change color .............................................. 30
   7.6.2 Sort on columns ........................................... 30
   7.6.3 Filter .......................................................... 31
   7.6.4 Save RTM-layout ........................................ 31
   7.6.5 Reset RTM-layout ...................................... 31
   7.7 Tree-view ....................................................... 31
   7.7.1 Change color .............................................. 31
   7.7.2 Tree representation ..................................... 31
   7.7.3 Multiple selection ...................................... 31
8 Extensibility requirements ............................................. 32
   8.1 TMF extension ................................................. 32
   8.2 NBI plug-in support ........................................ 32
   8.3 Internal database ............................................. 32
   8.4 Database populator .......................................... 32
9 Reliability ............................................................. 32
   9.1 Robustness ..................................................... 32
10 Quality requirements ................................................ 33
   10.1 Realistic data ................................................ 33
   10.2 Usability ....................................................... 33
   10.3 Understandability ......................................... 33
   10.4 Resource consumption .................................. 33
11 Maintainability ........................................................ 34
   11.1 Encapsulation ............................................... 34
   11.2 Consistency .................................................. 34
   11.3 Conciseness ............................................... 34
   11.4 Simplicity ...................................................... 34
12 Summary ............................................................. 35
1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification is to provide a description of the TMF814 Simulator’s functionality, and to serve as a product validation check. It will also deepen the understanding and stand as ground for rest of this project. Under the corresponding subheadings, an explanation of the functionality will be provided, along with requirements that need to be fulfilled such as data required for input and processing.

1.2 Document Conventions

In this document, every requirement statement has its own priority, which is rated low, medium or high. Also a time estimation on each requirement which helps when determining how much can be done in each iteration.

1.3 Intended Audience and Reading Suggestions

This document is mainly intended for Ericsson Solution Integrator, Solution Architect and Support Engineers to understand the functionality of the software. Begin with the overview sections and proceed through the sections that are most pertinent to you as a reader.

1.4 Stakeholders

Requirements will be based on the expectations and needs of the stakeholders of the software. The following are potential stakeholders:

- GDSC OSS Integration
- Interface implementers/Developers
- Other Ericsson employees
- Clients to Ericsson

The requirements specification will be created to comply with the needs of GDSC OSS Integration, as they will be the primary consumer of this product. To allow for future maintenance, additional functionality and adaptors, the needs of future interface implementers and developers will be considered.
1.5 Definitions and Abbreviations

GUI  Graphical User Interface  
OO   Object-Oriented  
SRS  Software Requirements Specification  
CORBA  Common Object Request Broker Architecture  
TMF  TeleManagement Forum  
MTNM  Multi Technology Network Management  
TMF814  Protocol using CORBA specialized for MTNM communication.  
TP   Termination Point  
TL   Topological Link, is primarily an administrative object used to convey a relationship between two TPs.  
MLSN  Multi Layer SubNetwork  
TCA  Threshold Crossing Alert  
MLRA  Multi Layer Routing Area  
ASAP  Alarm Severity Assignment Profile  
EM   Element Manager  
EMS  Element Management System
2 Overall Description

2.1 Product Perspective

The GSDC OSS-integrations team works with customizations, integration services and solutions design of Network Management Systems. One of these systems is ServiceOn and often, a solution specific script will be created or an adjustment in the system will be made for the client. In order to verify the expected outcome, tests need to be performed on a physical network which is very expensive to have solely for testing purposes. This often means that tests are performed on-site at clients. The objective of this project is to create a standalone virtual network simulator that communicates northbound over TMF814. This product will simulate an optical network that can be interacted with through a GUI as well as through TMF814. The product is intended to be used for testing purposes, and will provide a way to test integrations in-house.

2.2 Operating Environment

The typical workstations the GSDC OSS Integration has are normal laptops. All the laptops have Operating system Windows Vista, this will be the working environment and also the aim for simulator, although the simulator will be platform independent. The laptop will typically have one or two gigabyte primary memory and a dual core processor with at least 1.5 GHz capacity.

A local database is needed for the simulator to store the information needed, how to set it up will be included in the installation guide. By having a local database to test against the simulator can run on the laptop even without a real network connection.

2.3 Design and Implementation Constraints

The simulator and all the plug-ins should be written in Java. This will run on all the computers used by GSDC Integrations and make it possible to continue developing the simulator by GSDC personal even after this project has ended.

An open source database will be used to handle the possible big amount of data. The choice has fallen on MySQL because it’s free, works well with java and makes the setup on each computer simple.

2.4 Documentation

Deliverables include:

- JavaDoc
• A user manual
• Design Document
• Test Specification
• Thesis (in depth description and analysis, as well as results)
3 Overview of the system

The program will be built according to the Model-View-Control architecture (MVC). The model will contain the network data of the simulated OSS and encapsulates an internal database. The view will display all the information in the model, showing network elements, their correlation to each other and specific information about each network element. How the information is displayed depends on the selection of network elements and the functions that are implemented and loaded. The controller receives input and makes calls to the model. Because the controller handles all the logic, which is preferably kept modular, it will consist of multiple classes. A main controller will for instance handle general logic such as the menu system, while specific windows will be able to communicate with their own controller class. Multiple northbound interfaces will be able to be loaded at startup, given that they follow the given NBI.

3.1 System components

- Model
- View
- Control
- Interface
- Database
- TMF814 implementation of NBI

3.2 System dependencies

The program is divided into the components above, where the model is the central part. A graphical user interface that manipulates the model through views and controllers exists locally. The model can also be changed and interacted with through the northbound interface. The local GUI and the NBI should have the same functionality, though there might be some limitations in the NBI depending on the choice of protocol. If a new NBI is added to the simulator, functionality might have to be added to the GUI accordingly. The data in the database can be manipulated through either the GUI or the NBI.
4 Functional requirements

4.1 Model

4.1.1 Database storage

Priority: High
Iteration: 1-2
Ericsson mandatory: No
Time estimate: 100

Description: The network representation, along with network elements and all the at least the attributes required by the TMF814 functions should be stored in a local database.

4.1.2 Save

Priority: Medium
Iteration: 3
Ericsson mandatory: No
Time estimate: 3

Description: It should be possible to store the status of the different objects and the content of the database to a file that should be able to start on another computer running the simulator.

4.1.3 Load

Priority: Medium
Iteration: 3
Ericsson mandatory: No
Time estimate: 3

Description: Connected to 4.1.2. The previous saved files from the simulator should be able to load the same state properly so that the simulator-state is exactly as when saved. This means clearing the database and populating it with the correct data.

4.1.4 New project

Priority: High
Iteration: 2
Ericsson mandatory: No
Time estimate: 2

Description: Creating a new project should clear the database and prepare it for a new project. It should also clear any values on variables in the model.
4.2 Heart beat

4.2.1 Heart beat behavior

Priority: High  
Ericsson mandatory: Yes  
Iteration: 2  
Time estimate: 6

Description: The simulator should be able to reproduce the behavior of the heart beat function in ServiceOn systems. Internal alarms, or system alarms, should be raised for each node that is simulated not to have working heart beat.

4.2.2 Enable heart beat

Priority: High  
Ericsson mandatory: Yes  
Iteration: 2  
Time estimate: 1

Description: It should be possible to enable the heart beat functionality on each network element. The default value for each network element should be that heart beat is enabled. Enabling heart beat is done from the simulators point of view and not from NBI.

4.2.3 Disable heart beat

Priority: High  
Ericsson mandatory: Yes  
Iteration: 2  
Time estimate: 1

Description: It should be possible to disable the heart beat functionality for each network element. If heart beat is disabled for a network element an internal alarm is raised for this element.

4.3 Notification

Priority: High  
Ericsson mandatory: Yes  
Iteration: 1  
Time estimation: 24

Description: It should be possible to send messages (alarms) spontaneous through the NBI interface, where the simulator acts as a client instead of a server as will be the case for all the rest of the functionality.

4.4 Active alarm list

Priority: High  
Ericsson mandatory: Yes  
Iteration: 1  
Time estimation: 16
Description: The possible to answer requests from the NBI. This means that the simulator should react as a server. Exactly how it should react depends on the exact implementation of the NBI, and for the TMF814 case it will be discussed during chapter 5.

5 TMF 814

5.1 Connect to GUI Client

Priority: High  Ericsson Mandatory: Yes
Iteration: 1  Time estimate: 40

Description: It should be possible for the GUI Client to connect to the simulator server.

5.2 EMS manager

5.2.1 Function: getAllEMSAndMEActiveAlarms

Priority: High  Ericsson mandatory: Yes
Iteration: 2  Time estimate: 6

Description: When this function is used by a TMF-client the simulator should return all the alarm associated to this EMS. Both alarms from the network elements and internal alarm that the simulator has created by itself. Two excluding lists are used two filter on probable cause and severity. Matches against these lists are not to be returned. One list and one iterator for giving next list, if length of result is larger than the given maximum length, are return.

5.2.2 Function: getAllEMSSystemActiveAlarms

Priority: High  Ericsson mandatory: Yes
Iteration: 2  Time estimate: 6

Description: Related to requirement 5.2.1. Instead of returning alarms that don’t match the excluding lists from network elements and internal alarms, only the internal alarms are returned that are associated to this specific EMS.

5.2.3 Function: getAllTopLevelSubnetworks

Priority: High  Ericsson mandatory: No
Iteration: 2  Time estimate: 5
**Description:** This should return all the subnetworks that is associated with the specific EMS. It should return the whole subnetwork-object. Return as a list of subnetworks with the possibility of getting also an iterator depending on have big the result is.

5.2.4 Function: `getAllTopLevelSubnetworkNames`

<table>
<thead>
<tr>
<th>Priority</th>
<th>Ericsson mandatory</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>No</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Related to requirement 5.2.3. Instead of object it only returns the names on the objects. Return as list with the possibility of an additional iterator.

5.2.5 Function: `getAllTopLevelTopologicalLinks`

<table>
<thead>
<tr>
<th>Priority</th>
<th>Ericsson mandatory</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Yes</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Returns all Topological links between MLSN associated with the EMS.

5.2.6 Function: `getAllTopLevelTopologicalLinkNames`

<table>
<thead>
<tr>
<th>Priority</th>
<th>Ericsson mandatory</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Yes</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Related to 5.2.5. Returns only the name of each TL in the result instead of the whole TL-object.

5.2.7 Function: `getTopLevelTopologicalLink`

<table>
<thead>
<tr>
<th>Priority</th>
<th>Ericsson mandatory</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Description:** Should return the specific TL-object that is associated with the name of a TL, given as input.

5.2.8 Function: `getEMS`

<table>
<thead>
<tr>
<th>Priority</th>
<th>Ericsson mandatory</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium</td>
<td>No</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Description: This function should return all the stored information on the specific EMS. Information should follow the standard and contain for example name, version, owner, type and additional information.

5.2.9 Function: acknowledgeAlarms

Priority: High
Iteration: 2
Ericsson mandatory: Yes
Time estimate: 5

Description: This function will change the status of the given alarms to acknowledged. This goes for all alarms and TCA:s in the list.

5.2.10 Function: createTopologicalLink

Priority: High
Iteration: 2
Ericsson mandatory: No
Time estimate: 8

Description: This creates a TL or a MLSN according to the given topological data.

5.2.11 Function: deleteTopologicalLink

Priority: High
Iteration: 2
Ericsson mandatory: No
Time estimate: 2

Description: This function should remove a topological link, matching should be done via name.

5.2.12 Function: getAllEMSAndMEUnacknowledgedActiveAlarms

Priority: High
Iteration: 2
Ericsson mandatory: Yes
Time estimate: 3

Description: This function is connected to the 5.2.1. The result is filtered once before, making sure that only unacknowledged alarms is in the result.

5.2.13 Function: getAllEMSSystemUnacknowledgedActiveAlarms

Priority: High
Iteration: 2
Ericsson mandatory: Yes
Time estimate: 1

Description: Related to 5.2.2. Acknowledged alarms are filtered out before returning result.
5.2.14 Function: unacknowledgeAlarms

**Priority:** High  **Ericsson mandatory:** Yes  **Iteration:** 2  **Time estimate:** 2

**Description:** This functionality means that the alarms in the given list should change state to unacknowledged.

5.3 ManagedElement manager

5.3.1 Function: getManagedElement

**Priority:** High  **Ericsson mandatory:** Yes  **Iteration:** 2  **Time estimate:** 7

**Description:** Should retrieve the entire object structures of the managed element for the given managed element name.

5.3.2 Function: getAllManagedElements

**Priority:** High  **Ericsson mandatory:** Yes  **Iteration:** 2  **Time estimate:** 2

**Description:** Should retrieve the entire object structures of all of the Managed Elements.

5.3.3 Function: getAllManagedElementNames

**Priority:** High  **Ericsson mandatory:** Yes  **Iteration:** 2  **Time estimate:** 1

**Description:** Should retrieve the names of all of the Managed Elements. Related to getAllManagedElements, 5.3.2.

5.3.4 Function: getAllActiveAlarms

**Priority:** HIGH  **Ericsson mandatory:** Yes  **Iteration:** 2  **Time estimate:** 3

**Description:** Should retrieve all of the active alarms and TCAs for the specified managed element. Alarms that are not reported by the ME to the EMS should not be reported by this operation. Also, alarms which ASAP assigned severity is “NOTALARMED” should not be reported by this operation.
5.3.5 Function: getAllUnacknowledgedActiveAlarms

**Priority:** High

**Ericsson mandatory:** Yes

**Iteration:** 2

**Time estimate:** 2

**Description:** Should retrieve all of the active alarms and TCAs (for the specified managed element) that have not been acknowledged. Alarms that are not reported by the ME to the EMS should not be reported by this operation. Also, alarms which ASAP assigned severity is "NOTALARMED" should not be reported by this operation.

5.3.6 Function: getAllPTPs

**Priority:** Medium

**Ericsson mandatory:** No

**Iteration:** 2

**Time estimate:** 5

**Description:** Should retrieve the entire object structure of all of the PTPs and FTPs on the specified Managed Element that contain one or more of the NMS-specified PTP/FTP layer rates and that are capable of supporting one or more of the NMS-specified connection layer rates. If there are no PTPs/FTPs that match the layer constraints, an empty list is returned. A PTP/FTP will be returned regardless of connectivity to other managed elements and regardless of position in the subnetwork (both edgepoints of the subnetwork and the PTPs/FTPs that are internal to the subnetwork are reported).

5.3.7 Function: getAllPTPNames

**Priority:** Medium

**Ericsson mandatory:** No

**Iteration:** 2

**Time estimate:** 1

**Description:** This operation should have the same behavior as getAllPTPs(), but instead of returning the entire object structures, this operation returns their names.

5.3.8 Function: getAllFTPs

**Priority:** Medium

**Ericsson mandatory:** No

**Iteration:** 2

**Time estimate:** 2

**Description:** This operation has exactly the same behaviour as getAllPTPs(), but instead of returning both PTPs and FTPs it returns solely FTPs.

5.3.9 Function: getAllFTPNames

**Priority:** Med

**Ericsson mandatory:** No
5.3.10 Function: getAllPTPsWithoutFTPs

**Priority**: Med  
**Iterate**: 2  
**Time estimate**: 2

**Description**: This operation has exactly the same behavior as getAllPTPs(), but instead of returning both PTPs and FTPs it returns solely PTPs.

5.3.11 Function: getAllPTPNamesWithoutFTPs

**Priority**: Med  
**Iterate**: 2  
**Time estimate**: 1

**Description**: This operation should have the same behavior as getAllPTPsWithoutFTPs(), but instead of returning the entire object structures, this operation returns their names.

5.3.12 Function: getContainedCurrentTPs

**Priority**: low  
**Iterate**: 2  
**Time estimate**: 4

**Description**: Should retrieve the "current" CTPs that are contained within a given PTP, FTP or CTP, at specific layer rates. A "current" CTP is defined as a CTP that is either cross-connectable or cross-connected in the current mapping configuration.

5.3.13 Function: getContainedCurrentTPNames

**Priority**: low  
**Iterate**: 2  
**Time estimate**: 1

**Description**: This operation should have the same behavior as getContainedCurrentTPs(), but instead of returning the entire object structures, this operation returns their names.

5.3.14 Function: getContainedInUseTPs

**Priority**: med  
**Time estimate**: No

**Description**: This operation should have the same behavior as getContainedCurrentTPs(), but instead of returning the entire object structures, this operation returns their names.
Iteration: 2  Time estimate: 4

Description: Should retrieve the "in use" CTPs that are contained within a specific PTP, FTP or CTP, at specific layer rates. An "in use" CTP is defined as a CTP that is used by an SNC in any state (including pending), either as a CM end point or as an intermediate connection point, or a CTP that is terminated and mapped.

5.3.15 Function: getContainedInUseTPNames

Priority: med  Ericsson mandatory: No
Iteration: 2  Time estimate: 1

Description: This operation should have the same behavior as getContainedInUseTPs(), but instead of returning the entire object structures, this operation returns their names.

5.3.16 Function: getContainedPotentialTPs

Priority: low  Ericsson mandatory: No
Iteration: 2  Time estimate: 5

Description: Retrieves all of the CTPs for a given TP that it is potentially capable of supporting in all possible mapping configurations, at the specified rates and that are contained by the specified termination point. The TP may be a PTP, an FTP or a CTP. If the layerRateList is empty then contained CTPs at all flexible and/or static LayerRates are returned.

5.3.17 Function: getContainedPotentialTPNames

Priority: low  Ericsson mandatory: No
Iteration: 2  Time estimate: 1

Description: This operation should have the same behavior as getContainedPotentialTPs(), but instead of returning the entire object structures, this operation returns their names.

5.3.18 Function: getContainingTPs

Priority: med  Ericsson mandatory: No
Iteration: 2  Time estimate: 6

Description: Retrieves a list of the containing TPs given a CTP. This should return an PTP or FTP where there is only one level of containment. In a case of deeper containment this should return a list of CTPs and a PTP or FTP.
5.3.19 Function: getContainingTPNames

Priority: med  
Ericsson mandatory: No
Iteration: 2  
Time estimate: 2

Description: This operation should have the same behavior as getContainingTPNames(), but instead of returning the entire object structures, this operation returns their names.

5.3.20 Function: getContainingSubnetworkNames

Priority: med  
Ericsson mandatory: No
Iteration: 2  
Time estimate: 3

Description: Should return the list of subnetwork names that the Managed Element supplied as an input parameter belongs to.

5.3.21 Function: getTP

Priority: med  
Ericsson mandatory: No
Iteration: 2  
Time estimate: 5

Description: returns the termination point structure for the given TP name (CTP, FTP or PTP). The termination point name must be explicit (a generic endpoint specification may not be used in this case). The termination point structure contains SD1-16 transmission parameters. The transmission parameters returned will be the parameters in place on the actual termination point on the NE. If there are no transmission parameters or the TP does not actually exist on the NE, then transmissionParams will be empty. The field transmissionParams will also be empty for "potential" ATM VP/VC CTPs.

5.3.22 Function: setTPData

Priority: med  
Ericsson mandatory: No
Iteration: 2  
Time estimate: 16

Description: This service should allow the TMF GUI to set parameters on a specified Termination Point (CTP, PTP or FTP). This operation is best effort (except where specified otherwise for a particular parameter). The results of the operation are returned so that the NMS is aware of what modifications succeeded. If the source TP of a broadcast system is used as input, then the entire multipoint system will be affected based on the new parameter values for the source TP.
The `tpMappingMode` may be set with this operation. When the mode is set to `TM_TERMINATED_AND_AVAILABLE_FOR_MAPPING`, the EMS will terminate the specified CTP. In this case the EMS must create the specified CTP on the NE if it does not actually exist. Setting the `tpMappingMode` of an ATM VP or VC CTP can only be done if the CTP has been created. Note that the `tpMappingMode` can be set only on the ingress and egress CTPs of an ATM SNC since they are the only ones which may not be cross-connected.

No change to `tpMappingMode` or `trafficDescriptors` will take place if there is any active cross connect (NE cross connect) using the CTP passed in parameter.

The `transmissionParams` is a "delta" list that needs to be applied to the specified TP, i.e. only a subset of the parameters may be specified in the list, in which case only those should be applied in the NE. Transmission parameters are used to associate a TCA profile with a TP using this service. In this case there are potentially additional failure modes (see exceptions).

The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) as egress or ingress TMD by using the TMD's name amounts to an overwriting of the layered transmission parameters of the TP by the layered transmission parameters of the TMD, and to an overwriting of the additional info parameters of the TP by the "additional TP information" parameters of the TMD.

### 5.3.23 Function: createGTP

**Priority:** low  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 10

**Description:** TMF GUI should be able to use this operation to request the creation of a GTP. The it can either specify the list of CTPs comprising the GTP (the list is placed in the `listOfTPs` parameter) or, in the case of contiguous CTPs of the same layerRate, the NMS may list the first CTP (in the `initialCTPname` parameter) and the number of following CTPs (in the `numberOfCTPs` parameter). The NMS may also provide the EMS with the flexibility to decide the list of CTPs by using the `gtpEffort` parameter. In this flexible mode the EMS uses the `listOfTPs` parameter or the `initialCTPname/numberOfCTPs` parameters to determine the total requested bandwidth but it may return a GTP (newGTP parameter) with a different set of CTPs than those indicated in the original `createGTP` request. Note that this mode allows for the GTP components to be instantiated at a later time by the ME (e.g., upon detection of user's signal). Therefore the operation may successfully return a newGTP with an empty `listOfTPs` attribute (to be updated at a later time once the component CTPs are created in the ME).

### 5.3.24 Function: deleteGTP
### 5.3.25 Function: getGTP

**Priority:** low  
**Iteration:** 2  
**Ericsson mandatory:** No  
**Time estimate:** 4

**Description:** This operation should delete a GTP. An attempt to delete a GTP that is involved in a cross connection or SNC should be rejected by the EMS. This operation should be idempotent. If the service is called with the name of a non-existent GTP, it should succeed.

### 5.3.26 Function: getAllGTPs

**Priority:** low  
**Iteration:** 2  
**Ericsson mandatory:** No  
**Time estimate:** 3

**Description:** Should return the GTP structure for the given GTP name

### 5.3.27 Function: getAllGTPNames

**Priority:** low  
**Iteration:** 2  
**Ericsson mandatory:** No  
**Time estimate:** 1

**Description:** This operation should have the same behavior as getAllGTPs(), but instead of returning the entire object structures, this operation returns their names.

### 5.3.28 Function: getContainingGTP

**Priority:** low  
**Iteration:** 2  
**Ericsson mandatory:** No  
**Time estimate:** 4

**Description:** Should return the name of the GTP containing a given CTP. If the CTP is not contained in a GTP, the gtp output parameter is left empty.

### 5.3.29 Function: modifyGTP

**Priority:** low  
**Iteration:** 2  
**Ericsson mandatory:** No  
**Time estimate:** 10

**Description:** The modify GTP operation is used to add TPs to or delete TPs from a GTP. For a given request, the NMS can only add or delete CTPs, not
both. It is not possible to add a CTP that is already involved in a cross connection or SNC, or that is part of another GTP. Attempts to modify a GTP that is involved in a cross connection (or SNC) should be rejected by the EMS. The operation is best-effort, i.e., the EMS will add or delete as many of the identified CTPs as possible. If the service is called with the name of a non-existent GTP or CTP, it will fail. If the NMS created a GTP with gtpEffort equal to EFFORTSAME, this also implies that the EMS will not modify on its own, i.e., the EMS will only modify the GTP if requested by the NMS via the modifyGTP operation.

5.3.30 Function: getPotentialFixedCCs

Description: Should retrieve fixed connection schemes related to normal and inverse multiplexing. A cross connection is identified as a fixed SNC using additional information.

The operation takes as an input a TP and if this TP may be or is involved in a multiplexing or inverse multiplexing scheme, it will report:

- The TP containing the fixed layer.
- The list of potential fixed cross connects that will exist if the containing TP's clientConnectivity or serverConnectivity is set to "Connected" (i.e. If the containing TP is set to multiplexing or inverse multiplexing).

If the clientConnectivity (resp. serverConnectivity) of the TP is currently set to "Connected", the potentialCCList matches the list of active fixed cross connects that involve its client TPs (resp. server TPs).

If the TP client layer (resp. server layer) is always fixed cross connected (hard-wired), the potentialCCList always matches the list of active fixed cross connects that involve the client TPs (resp. server TPs).

5.3.31 Function: getAllCrossConnections

Description: Should retrieve a list of the cross-connects for the specified managed element at the specified layer rates. This operation returns cross-connects between CTPs/FTPs and between GTPs.

5.3.32 Function: getAllFixedCrossConnections

Description: Should retrieve a list of the cross-connects for the specified managed element at the specified layer rates. This operation returns cross-connects between CTPs/FTPs and between GTPs.
Description: Should have the exact same behaviour as getAllCrossConnections, but returns only fixed cross connection object structures.

5.3.33 Function: setGtpAlarmReportingOn

Priority: Low
Iteration: 2

Ericsson mandatory: No
Time estimate: 5

Description: Should enable (activate, or turn on) alarm reporting for a GTP. The EMS sends an attribute value change notification in case of success.

5.3.34 Function: setGtpAlarmReportingOff

Priority: Low
Iteration: 2

Ericsson mandatory: No
Time estimate: 1

Description: Should disable (deactivate, or turn off) alarm reporting for a GTP. The EMS sends an attribute value change notification in case of success.

5.3.35 Function: verifyT DMAssignment

Priority: Medium
Iteration: 2

Ericsson mandatory: No
Time estimate: 6

Description: This service verifies the egress and/or ingress TMD state of the specified TP identified by tpName.

According to the requested transmission direction it checks whether a TMD is assigned as egress and/or ingress TMD to the TP. If this is true it first verifies the existence of the assigned TMD(s) and then compares the transmission parameters of the TMD and TP, and the additional TP info parameters of the TMD with the additional info parameters of the TP.

If a TMD parameter is not present as a TP parameter or is present but with a different value, this is called a "parameter mismatch", and the TMD state will take the value "mismatch". The service returns the current TMD state and all mismatched transmission or additional TP info parameters of the TMD (which should mostly be none) as its output.
5.4 Equipment

5.4.1 Function: getAllEquipment

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 4

**Description:** Should return all the information from the Managed Element or the Equipment holder. It should match on the name, which is given. Returns a list on Equipments or Holders that has a maximum length of how_many. If more things exists an iterator should be able to give the rest.

5.4.2 Function: getAllEquipmentNames

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 1

**Description:** Related to 5.4.1. Instead of returning the whole object only the names of all equipment should be returned.

5.4.3 Function: getAllSupportedPTPs

**Priority:** med  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 4

**Description:** This should return all the PTPs/FTPs that are connected, directly, to the given equipment. PTPs returned share their physical layer with the given equipment, e.g. the port is physical on the equipment. Could be a port, a connected fiber, connected wire etc. The FTPs that are returned are those which are implemented by the physical circuitry supported by the equipment and will include FTPs that are named from the specified equipment. Returned as a list and an iterator, depending on how_many variable.

5.4.4 Function: getAllSupportedPTPNames

**Priority:** med  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 1

**Description:** Connected to 5.4.3. Should only return a list on name instead of actual objects.

5.4.5 Function: getAllSupportingEquipment

**Priority:** med  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 5

**Description:** Return the equipments and the equipment holders that is supported by a specific equipment holder.
5.4.6 Function: getAllSupportingEquipmentNames

Priority: med  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 1

Description: Related to 5.4.5. Returns only the names instead of a list of actual equipment or holders.

5.4.7 Function: getContainedEquipment

Priority: med  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 2

Description: Returns the equipments and the equipment holders that is seated on a specific equipment holder. This differs from the getAllEquipment, 5.4.1, in the way that it ONLY looks at the next level in the containment hierarchy. As input it takes the name of the equipment holder.

5.4.8 Function: getEquipment

Priority: med  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 2

Description: Returns the equipment or holder given the name of it.

5.4.9 Function: provisionEquipment

Priority: med  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 16

Description: This operation allows the NMS to permanently provision equipment in an equipment holder in a Managed Element. The successful result of this operation is the creation or update of the equipment object and the creation of all its related objects such as Termination Points. The resulting object will be returned. It requires that correct equipmentCreateData is given.

5.4.10 Function: setAlarmReportingOff

Priority: low  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 2

Description: This should disable the functionality for an equipment or holder to report alarms. This do not effect TP alarms. EMS sends an attribute value change notification in case of success.

5.4.11 Function: setAlarmReportingOn

Priority: low  
Iteration: 2  
Ericsson mandatory: No  
Time estimate: 1
Description: Related to 5.4.11. This function should turn on the possibility for an equipment to report alarms. It only turns on alarms from the equipment or holder, not for the TP. The EMS should send an attribute value change notification when successful.

5.4.12 Function: unprovisionEquipment

Priority: med
Iteration: 2
Ericsson mandatory: No
Time estimate: 6

Description: Related to 5.4.9. It should try to remove data instead of creating a new Equipment. It first tries to set the equipment to OUT_OF_SERVICE_BY_MAINTENANCE.

5.4.13 Function: getSupportedEquipment

Priority: med
Iteration: 2
Ericsson mandatory: No
Time estimate: 4

Description: This should return all the equipment that is dependent on the specific given equipment. Could be that a power pack supports several circuit packs and then the circuit packs should be returned.

5.4.14 Function: getSupportedEquipmentNames

Priority: med
Iteration: 2
Ericsson mandatory: No
Time estimate: 1

Description: Related to 5.4.13. This should only return the specific names instead of the whole object.

5.4.15 Function: getSupportingEquipment

Priority: med
Iteration: 2
Ericsson mandatory: No
Time estimate: 4

Description: Returns the equipment that supports a given piece of equipment. Could be power source for the equipment, e.g. equipment that needs to be there for the given equipment to work properly.

5.4.16 Function: getSupportingEquipmentNames

Priority: med
Iteration: 2
Ericsson mandatory: No
Time estimate: 1

Description: Related to 5.4.15. Should only return the names and not whole objects.
6 Nonfunctional requirements

6.1 ServiceOn resemblance

Priority: Medium  Ericsson mandatory: No
Iteration: 3  Time estimate: 4

Description: The simulator should to some extend resemble the ServiceOn system, when it comes to the GUI and the interactions that the simulator supports.

6.2 Nonfunctional requirements for TMF 814

7 GUI

Skapa grafik

7.1 Exit

Priority: High  Ericsson mandatory: No
Iteration: 1  Time estimate: 1

Description: It should be possible to close the simulator through the two standardized ways, through File->Exit and when clicking the x button in the upper-right corner.

7.2 Switching view

Priority: High  Ericsson mandatory: No
Iteration: 1  Time estimate: 2

Description: Changing between Map and RTM view should be possible.

7.3 Resizing

Priority: Low  Ericsson mandatory: No
Iteration: 3  Time estimate: 1

Description: It should be possible to resize the simulator GUI.
7.4 Menu

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 2  

**Description:** A menu containing standardized features as New, Save, Load, Exit, About, Settings should exist.

7.5 Map

7.5.1 Adding NE

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 4  

**Description:** By choosing the add NE in the toolbar it should be possible to click wherever the NE should be placed.

7.5.2 Moving NE

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 3  

**Description:** It should be possible to move a NE through drag and drop.

7.5.3 Removing NE

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 1  

**Description:** When having the remove tool selected it should be possible to remove a NE by clicking on it.

7.5.4 Adding TL

**Priority:** Med  
**Ericsson mandatory:** No  
**Iteration:** 1-2  
**Time estimate:** 4  

**Description:** It should be possible to add a topological link between two NE with the draw TL tool

7.5.5 Removing TL

**Priority:** med  
**Ericsson mandatory:** No  
**Iteration:** 1-2  
**Time estimate:** 2  

**Description:** It should be possible to remove a topological link by clicking on it, if the remove tool is selected.
### 7.5.6 More info on NE

**Priority:** med  
**Ericsson mandatory:** No  
**Iteration:** 2  
**Time estimate:** 8

**Description:** By right-clicking on the NE a context-menu with more info and options on the NE should be presented.

### 7.5.7 Showing alarms

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 2

**Description:** Alarms on a NE should be shown by changing the color of the NE.

### 7.5.8 Heart beat failure

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 2

**Description:** Heart beat failure should be indicated by adding a small lightning bolt on the NE.

### 7.6 RTM

#### 7.6.1 Change color

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 2

**Description:** Should change color depending on status.

#### 7.6.2 Sort on columns

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 1

**Description:** It should be possible to click on each column and sort on this specific column. Clicking ones should sort descending, click again and it should sort ascending.
### 7.6.3 Filter

**Priority:** Med  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 4

**Description:** It should be possible to filter on State, Severity, Type, Domain and time. Only alarms meeting the filters should be shown.

### 7.6.4 Save RTM-layout

**Priority:** Low  
**Ericsson mandatory:** No  
**Iteration:** 3  
**Time estimate:** 3

**Description:** Should be possible to save the configuration of the layout for RTM, this layout will automatically be loaded each time RTM is used.

### 7.6.5 Reset RTM-layout

**Priority:** Low  
**Ericsson mandatory:** No  
**Iteration:** 3  
**Time estimate:** 1

**Description:** If the layout for RTM is resettled, then it should return to a predefined state.

### 7.7 Tree-view

#### 7.7.1 Change color

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 3

**Description:** Should change color depending on status.

#### 7.7.2 Tree representation

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 4

**Description:** Sub-networks and network elements should visible in a tree, with possibility to filter Map and RTM with the tree.

#### 7.7.3 Multiple selection

**Priority:** High  
**Ericsson mandatory:** No  
**Iteration:** 1  
**Time estimate:** 1
Description: Should be possible to select multiple nodes, not only from the same subnetwork. If a subnetwork is selected then it is assumed that all the children of this node is also selected.

8 Extensibility requirements

8.1 TMF extension

Priority: High  Ericsson mandatory: No
Iteration: 1  Time estimate: 24

Description: It should be possible to add the missing pieces of the TMF814 standard to make a complete coverage after this project.

8.2 NBI plug-in support

Priority: High  Ericsson mandatory: No
Iteration: 1  Time estimate: 24

Description: Other north bound interfaces should be able to integrate plug-ins given that they follow the NBI-interface.

8.3 Internal database

Priority: Medium  Ericsson mandatory: No
Iteration: 3  Time estimate: 24

Description: Support for changing to another internal database should be given. The other database has to be implemented in a wrapper following the interface for internal database.

8.4 Database populator

Priority: Low  Ericsson mandatory: No
Iteration: 3  Time estimate: 24

Description: Different populators for different external sources should be able to populate the database, given that they follow an given interface.

9 Reliability

9.1 Robustness

Priority: High  Ericsson mandatory: No
Iteration: 3  Time estimate: 16
Description: The program should operate despite abnormalities in input, calculations, etc. Fuzz-testing should be used to prove robustness.

10 Quality requirements

10.1 Realistic data

Priority: Medium | Ericsson mandatory: No
Iteration: 2 | Time estimate: 16

Description: The database should contain realistic network data that the stakeholders find useful.

10.2 Usability

Priority: High | Ericsson mandatory: No
Iteration: 3 | Time estimate: 16

Description: The program should be practical to start-up and begin using. The user interface should be easy to understand for people with some knowledge when it comes to OSS.

10.3 Understandability

Priority: High | Ericsson mandatory: No
Iteration: 3 | Time estimate: 16

Description: Not only should the program itself be easy to use, but the purpose of the final product should be clear with the stakeholders as well. User documentation must therefore be clearly written, so that the value of this program is easily understandable.

10.4 Resource consumption

Priority: High | Ericsson mandatory: No
Iteration: 3 | Time estimate: 16

Description: The program should not consume an undesirable amount of memory or CPU.
11 Maintainability

11.1 Encapsulation

Priority: Medium  \hspace{1cm} \textbf{Ericsson mandatory: No}
Iteration: 3  \hspace{1cm} \textbf{Time estimate: 16}

Description: When possible, functionality should be encapsulated.

11.2 Consistency

Priority: Medium  \hspace{1cm} \textbf{Ericsson mandatory: No}
Iteration: 3  \hspace{1cm} \textbf{Time estimate: 16}

Description: There should be consistency in indentations, comments and variables.

11.3 Conciseness

Priority: Medium  \hspace{1cm} \textbf{Ericsson mandatory: No}
Iteration: 3  \hspace{1cm} \textbf{Time estimate: 16}

Description: There should be a minimization of excessive or redundant information or processing.

11.4 Simplicity

Priority: Medium  \hspace{1cm} \textbf{Ericsson mandatory: No}
Iteration: 3  \hspace{1cm} \textbf{Time estimate: 16}

Description: Although optimality is desirable, it should not override clear, understandable code apart from if a function is run several times or effects the performance in a measurable way.
12 **Summary**

The total amount of man-hours calculated for the requirements given in this document are just a rough estimate. The actual time for each implementation including preparations for each functionality and each requirement could differ in both ways.

Total: 743 Man-Hours

Iteration 1: 262 Man-Hours

Iteration 2: 290 Man-Hours

Iteration 3: 191 Man-Hours

Total amount of Man-Hours that are assigned for designing and implementing to this project is around 800-850 depending on some holidays. Some of the features in iteration 3 will be parts of the development in earlier iterations so the time will somewhat shift from this iteration to the earlier iterations.

This time is not including testing, nor is it including the documentation writing at the end of the project.
<table>
<thead>
<tr>
<th>Prepared (also subject responsible if other)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAB/ZA/MNO Mikael Riedel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Approved</th>
<th>Checked</th>
<th>Date</th>
<th>Rev</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2010-03-17</td>
<td>PA1</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D – JavaDoc
# Table Of Content

<table>
<thead>
<tr>
<th>Class</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.ericsson.eos</td>
<td>5</td>
</tr>
<tr>
<td>Start</td>
<td>5</td>
</tr>
<tr>
<td>com.ericsson.eos.config</td>
<td>6</td>
</tr>
<tr>
<td>AlarmTypes</td>
<td>6</td>
</tr>
<tr>
<td>Config</td>
<td>7</td>
</tr>
<tr>
<td>com.ericsson.eos.controller</td>
<td>11</td>
</tr>
<tr>
<td>MainController</td>
<td>11</td>
</tr>
<tr>
<td>RTAMController</td>
<td>13</td>
</tr>
<tr>
<td>TPController</td>
<td>16</td>
</tr>
<tr>
<td>TreeController</td>
<td>17</td>
</tr>
<tr>
<td>com.ericsson.eos.database</td>
<td>20</td>
</tr>
<tr>
<td>MySQLConnector</td>
<td>20</td>
</tr>
<tr>
<td>MySQLDelete</td>
<td>21</td>
</tr>
<tr>
<td>MySQLGetters</td>
<td>24</td>
</tr>
<tr>
<td>MySQLInit</td>
<td>36</td>
</tr>
<tr>
<td>MySQLSetters</td>
<td>37</td>
</tr>
<tr>
<td>MySQLUpdater</td>
<td>50</td>
</tr>
<tr>
<td>XmlParser</td>
<td>58</td>
</tr>
<tr>
<td>com.ericsson.eos.debugger</td>
<td>61</td>
</tr>
<tr>
<td>Debugger</td>
<td>61</td>
</tr>
<tr>
<td>com.ericsson.eos.dynamicLoader</td>
<td>64</td>
</tr>
<tr>
<td>DynamicLoader</td>
<td>64</td>
</tr>
<tr>
<td>com.ericsson.eos.helper</td>
<td>66</td>
</tr>
<tr>
<td>InputChecker</td>
<td>66</td>
</tr>
<tr>
<td>Splitter</td>
<td>67</td>
</tr>
<tr>
<td>TreePathDivider</td>
<td>68</td>
</tr>
<tr>
<td>com.ericsson.eos.interfaces</td>
<td>70</td>
</tr>
<tr>
<td>DBI</td>
<td>70</td>
</tr>
<tr>
<td>DebugInterface</td>
<td>101</td>
</tr>
<tr>
<td>NBI</td>
<td>102</td>
</tr>
<tr>
<td>com.ericsson.eos.junitTests</td>
<td>116</td>
</tr>
<tr>
<td>AllTests</td>
<td>116</td>
</tr>
<tr>
<td>GetterTester</td>
<td>117</td>
</tr>
<tr>
<td>MySQLTest</td>
<td>117</td>
</tr>
<tr>
<td>com.ericsson.eos.model</td>
<td>119</td>
</tr>
<tr>
<td>Model</td>
<td>120</td>
</tr>
<tr>
<td>ModelAlarm</td>
<td>121</td>
</tr>
<tr>
<td>ModelCtp</td>
<td>127</td>
</tr>
</tbody>
</table>
ModelEms ....................................................................................................................... 133
ModelHelper ................................................................................................................... 135
ModellInit ....................................................................................................................... 140
ModelMe ......................................................................................................................... 141
ModelMlsn ...................................................................................................................... 145
ModelPtp ......................................................................................................................... 148
ModelSelection ............................................................................................................... 151
ModelTca ....................................................................................................................... 153
ModelTI .......................................................................................................................... 158
MyTableModel ................................................................................................................ 161
com.ericsson.eos.services ................................................................................................. 163
NameService .................................................................................................................. 163
NotificationService ......................................................................................................... 164
com.ericsson.eos.view ...................................................................................................... 165
AboutPanel ..................................................................................................................... 166
EMSPanel ....................................................................................................................... 167
MEPanel ......................................................................................................................... 168
MLSNPanel ..................................................................................................................... 170
MainView ....................................................................................................................... 171
ParseStatusPanel ........................................................................................................... 174
StatusPanel .................................................................................................................... 175
TPPanel .......................................................................................................................... 176
TPView ........................................................................................................................... 179
TopoLinkPanel ................................................................................................................ 182
TreeModelIconRenderer ................................................................................................. 184
TreeView ......................................................................................................................... 185
XkcdPanel ....................................................................................................................... 188
com.ericsson.eos.view.alarm ............................................................................................ 190
AlarmAlarmPanel ............................................................................................................ 190
AlarmTCAPanel .............................................................................................................. 194
NTAlarmView ................................................................................................................ 197
NTTCAView .................................................................................................................. 200
com.ericsson.eos.view.rtam ............................................................................................. 204
CustomTableCellRenderer ............................................................................................... 204
RTAMAlarm .................................................................................................................... 205
RTAMFilter .................................................................................................................... 208
RTAMTCA ...................................................................................................................... 209
RTAMView ..................................................................................................................... 212
com.ericsson.helper ........................................................................................................ 215
AlarmCreator .................................................................................................................. 215
NameExtractor ................................................................. 218
com.ericsson.nbilmp .................................................... 220
 NBilmp ................................................................. 220
 TestStarter ............................................................ 229
com.ericsson.poalmp .................................................. 230
 ASAPIteratore IPOAlmp ........................................... 233
 BackupIdIteratore IPOAlmp ..................................... 234
 CCIteratore IPOAlmp ............................................... 235
 CallAndTopLevelConnectionsAndSNCslterator IPOAlmp .................................................. 236
 CallAndTopLevelConnectionsIteratore IPOAlmp ........ 237
 Common IPOAlmp .................................................. 238
 CurrentMaintenanceOperationIteratore IPOAlmp .......... 239
 EMSMgr IPOAlmp .................................................. 240
 EProtectionGroupIteratore IPOAlmp ......................... 248
 EmsSessionFactory IPOAlmp ...................................... 249
 EmsSession_IPOAlmp ............................................. 250
 EquipmentInventoryMgr IPOAlmp ............................ 252
 EquipmentOrHolderIteratore IPOAlmp ....................... 257
 EventIteratore IPOAlmp ........................................... 258
 FDFrIteratore IPOAlmp ........................................... 259
 FDIteratore IPOAlmp .............................................. 260
 FlowDomainMgr IPOAlmp ........................................ 261
 GTPIteratore IPOAlmp ............................................ 270
 GuiCutThroughMgr IPOAlmp ...................................... 271
 MDFrIteratore IPOAlmp ........................................... 273
 MDFIteratore IPOAlmp ............................................. 274
 MLSNPPlIteratore IPOAlmp ....................................... 275
 MLSNPPLinkIteratore IPOAlmp .................................. 276
 MLSNPPLinkMgr IPOAlmp ......................................... 277
 MLSNPPPMgr IPOAlmp ............................................ 280
 MaintenanceMgr IPOAlmp ......................................... 282
 ManagedElementIteratore IPOAlmp ......................... 284
 ManagedElementMgr IPOAlmp .................................. 285
 MultiLayerSubnetworkMgr IPOAlmp ......................... 294
 NamingAttributesIteratore IPOAlmp ......................... 312
 NmsSession IPOAlmp ............................................. 313
 PMDataIteratore IPOAlmp ......................................... 315
 PMPIteratore IPOAlmp ............................................ 316
 PerformanceManagementMgr IPOAlmp ...................... 317
 ProtectionGroupIteratore IPOAlmp ............................ 323
Package com.ericsson.eos

Class Summary

Start

The Class Start.

com.ericsson.eos

Class Start

defined in java.lang.Object

+--com.ericsson.eos.Start

< Constructors > < Methods >

public class Start
extends java.lang.Object

The Class Start. This is the Entry-point for this simulator and all the including functionality.

Constructors

Start

public Start(java.lang.String arg)

Instantiates a new instance of the simulator.

Parameters:

arg - Parameter to tell which mode the program should start in, "install", "parse" or ""

Methods

main

public static void main(java.lang.String[] args)

The main method.

Parameters:

args - the arguments
The Class AlarmTypes. This is a class for handling templates for alarms, stored in AlarmTypes.xml. It uses the same parser as the XmlParser class.

Constructors

AlarmTypes

public AlarmTypes(Model model)

Instantiates a new instance of this class. It tries to read from the alarm-files as soon as it is created.

Parameters:

model - the model

Methods

getAlarmTypes

public java.util.ArrayList getAlarmTypes()

Gets the list of alarm types. This list has been parsed from the xml-file.

Returns:

the alarm types
setup

public void setup()

Setup. This function sets the parser and starts the parsing.

com.ericsson.eos.config

Class Config

java.lang.Object

  --> java.util.Observable

    --> com.ericsson.eos.config.Config

All Implemented Interfaces:
  java.io.Serializable

< Constructors > < Methods >

public class Config
extends java.util.Observable
implements java.io.Serializable

The Class Config. This class handles the reading and storing of configuration data. It stores the
dedbuggLevel, table layouts, path to log-file and which mode to use at startup. This is where information
that needs to be stored that hasn't anything to do with the network should be stored. The config.conf,
which is the file used, is easy to alter with a normal text-editor.

Constructors

Config

public Config()

    Instantiates a new Config-object. Default path to conf-file is config.conf Uses default values, but
tries to read values from the config-file, if other values are found they will be updated.

Config

public Config(java.lang.String file)

    Instantiates a new config with a given path.

Parameters:

  file - the path to the configuration file.
**getDebugFile**

public java.lang.String getDebugFile()

    Gets the debug file.
    Returns:
        the debug file

---

**getDebugLevel**

public int getDebugLevel()

    Gets the debug level.
    Returns:
        the debug level

---

**getMode**

public java.lang.String getMode()

    Gets the mode.
    Returns:
        the mode

---

**getNT_AlarmSortOrder**

public java.lang.String[] getNT_AlarmSortOrder()

    Gets the n t_ alarm sort order.
    Returns:
        the n t_ alarm sort order

---

**getNT_TCASortOrder**

public java.lang.String[] getNT_TCASortOrder()

    Gets the n t_ tca sort order.
    Returns:
        the n t_ tca sort order
**getPath**

```java
public java.lang.String getPath()
```

Gets the path.

**Returns:**

the path

---

**reSetConfig**

```java
public void reSetConfig(boolean b)
```

Resets the config-file. Alternatively it stores the stored data to the file. If the parameter is true then the values will be reset to default before storing the information.

**Parameters:**

- b - the b

---

**setDebugFile**

```java
public void setDebugFile(java.lang.String path)
```

Sets the debug file.

**Parameters:**

- path - the new debug file

---

**setDebugLevel**

```java
public void setDebugLevel(int debugLevel)
```

Sets the debug level.

**Parameters:**

- debugLevel - the new debug level

---

**setMode**

```java
public void setMode(java.lang.String str)
```

Sets the mode.

**Parameters:**

- str - the new mode
setNT_AlarmSortOrder

public void setNT_AlarmSortOrder(java.lang.String[] list)
    Sets the n t_ alarm sort order.
    Parameters:
        list - the new n t_ alarm sort order

setNT_TCASortOrder

public void setNT_TCASortOrder(java.lang.String[] list)
    Sets the n t_ tca sort order.
    Parameters:
        list - the new n t_ tca sort order

setPath

public void setPath(java.lang.String path)
    Sets the path.
    Parameters:
        path - the new path
Class Summary

**MainController**
The Class MainController.

**RTAMController**
The Class RTAMController.

**TPController**
The Class TPController.

**TreeController**
The Class TreeController.

---

com.ericsson.eos.controller

Class MainController

java.lang.Object

```plaintext
   +--com.ericsson.eos.controller.MainController
```

**All Implemented Interfaces:**
java.awt.event.ActionListener, java.awt.event.WindowListener, java.io.Serializable

< Constructors > < Methods >

public class MainController extends java.lang.Object implements java.awt.event.ActionListener, java.awt.event.WindowListener, java.io.Serializable

The Class MainController.

Constructors

**MainController**

```plaintext
public MainController(MainView mv)
```

Instantiates a new main controller.

**Parameters:**

mv - the mv

Methods
**actionPerformed**

public void actionPerformed(java.awt.event.ActionEvent ae)

---

**windowActivated**

public void windowActivated(java.awt.event.WindowEvent arg0)

---

**windowClosed**

public void windowClosed(java.awt.event.WindowEvent arg0)

---

**windowClosing**

public void windowClosing(java.awt.event.WindowEvent arg0)

---

**windowDeactivated**

public void windowDeactivated(java.awt.event.WindowEvent arg0)

---

**windowDeiconified**

public void windowDeiconified(java.awt.event.WindowEvent arg0)

---

**windowIconified**

public void windowIconified(java.awt.event.WindowEvent arg0)

---

**windowOpened**

public void windowOpened(java.awt.event.WindowEvent arg0)
Class RTAMController

java.lang.Object

---com.ericsson.eos.controller.RTAMController

All Implemented Interfaces:
    java.awt.event.ActionListener, java.awt.event.MouseListener,
    javax.swing.event.TableColumnModelListener

Constructors

public class RTAMController
extends java.lang.Object
implements java.awt.event.ActionListener, java.awt.event.MouseListener,
javax.swing.event.TableColumnModelListener

The Class RTAMController.

Constructors

RTAMController

public RTAMController(RTAMView view,
                       MainView mv)

    Instantiates a new RTAM controller. Needs the RTAMView and MainView as parameters to be
    able to update some of the status related stuff.

    Parameters:
    view - the view
    mv - the mv

Methods

actionPerformed

public void actionPerformed(java.awt.event.ActionEvent ae)

columnAdded

public void columnAdded(javax.swing.event.TableColumnModelEvent arg0)
columnMarginChanged
public void columnMarginChanged(javax.swing.event.ChangeEvent arg0)

--
columnMoved
public void columnMoved(javax.swing.event.TableColumnModelEvent arg0)

--
columnRemoved
public void columnRemoved(javax.swing.event.TableColumnModelEvent arg0)

--
columnSelectionChanged
public void columnSelectionChanged(javax.swing.event.ListSelectionEvent arg0)

--
filterActionPerformed
public void filterActionPerformed(java.awt.event.ActionEvent evt)
    Filter action performed. This creates a new Filter for the RTAMView.
    Parameters:
    evt - the evt

--
mouseClicked
public void mouseClicked(java.awt.event.MouseEvent me)

--
mouseEntered
public void mouseEntered(java.awt.event.MouseEvent arg0)
mouseExited
public void mouseExited(java.awt.event.MouseEvent arg0)

mousePressed
public void mousePressed(java.awt.event.MouseEvent e)

mouseReleased
public void mouseReleased(java.awt.event.MouseEvent e)

nt_AlarmActionPerformed
public void nt_AlarmActionPerformed(java.awt.event.ActionEvent evt)
    Nt_alarm action performed. Change to alarm-view.
    Parameters:
        evt - the evt

nt_TCAActionPerformed
public void nt_TCAActionPerformed(java.awt.event.ActionEvent evt)
    Nt_tca action performed. Change to TCA-view
    Parameters:
        evt - the evt

showAlarm
public void showAlarm(java.lang.String type,
                      java.lang.String notificationID)
    Show alarm. This function request the information from the model about a specific alarm or tca.
The information is extracted and displayed in the NT_AlarmPanel or the NT_TCAPanel.
    Parameters:
        type - the type
        notificationID - the notification id
com.ericsson.eos.controller

Class TPController

java.lang.Object
   +--com.ericsson.eos.controller.TPController

All Implemented Interfaces:
   java.awt.event.ActionListener, java.awt.event.MouseListener, java.io.Serializable,
   javax.swing.event.ListSelectionListener

< Constructors > < Methods >

public class TPController
extends java.lang.Object
implements java.awt.event.ActionListener, java.awt.event.MouseListener, java.io.Serializable,
javax.swing.event.ListSelectionListener

The Class TPController. This is the Controller for the Termination Point Main View.

Constructors

TPController

public TPController(TPView tp,
                     Model model,
                     MainView mv)

   Instantiates a new TP controller. Need the views that is should interact with and also the model to
being able to update the information showed.

   Parameters:
   tp - the TPView
   model - the Model
   mv - the MainView

Methods

actionPerformed

public void actionPerformed(java.awt.event.ActionEvent ae)

mouseClicked

public void mouseClicked(java.awt.event.MouseEvent me)
mouseEntered
public void mouseEntered(java.awt.event.MouseEvent arg0)

mouseExited
public void mouseExited(java.awt.event.MouseEvent arg0)

mousePressed
public void mousePressed(java.awt.event.MouseEvent e)

mouseReleased
public void mouseReleased(java.awt.event.MouseEvent e)

valueChanged
public void valueChanged(javax.swing.event.ListSelectionEvent lse)

com.ericsson.eos.controller

Class TreeController
java.lang.Object
   +--com.ericsson.eos.controller.TreeController

All Implemented Interfaces:
   java.awt.event.ActionListener, java.awt.event.MouseListener, java.io.Serializable,
   javax.swing.event.TreeSelectionListener

< Constructors > < Methods >

public class TreeController
   extends java.lang.Object
   implements java.awt.event.ActionListener, java.awt.event.MouseListener, java.io.Serializable,
             javax.swing.event.TreeSelectionListener

The Class TreeController. This is the controller class for the tree-structure to the left in the simulator.
Constructors

TreeController

public TreeController(TreeView tv, Model m, MainController m2, MainView mv, ModelSelection mts)

Instanitates a new tree controller.

Parameters:
  tv - the tv
  m - the m
  m2 - the m2
  mv - the mv
  mts - the mts

Methods

actionPerformed

public void actionPerformed(java.awt.event.ActionEvent ae)

addToCurrentNode

public void addToCurrentNode(java.lang.String name)

   Adds the to current node.

Parameters:
  name - the name

mouseClicked

public void mouseClicked(java.awt.event.MouseEvent me)

mouseEntered

public void mouseEntered(java.awt.event.MouseEvent arg0)
mouseExited

public void mouseExited(java.awt.event.MouseEvent arg0)

mousePressed

public void mousePressed(java.awt.event.MouseEvent e)

mouseReleased

public void mouseReleased(java.awt.event.MouseEvent e)

removeCurrentNode

public void removeCurrentNode()

    Removes the current node.

valueChanged

public void valueChanged(javax.swing.event.TreeSelectionEvent tse)
The Class MySQLConnector.

MySQLDelete
The Class MySQLDelete.

MySQLGetters
The Class MySQLGetters.

MySQLInit
The Class MySQLInit.

MySQLSetters
The Class MySQLSetters contains all the JDBC set-functions.

MySQLUpdater
The Class MySQLUpdater.

XmlParser
The Class XmlParser.

com.ericsson.eos.database

Class MySQLConnector

java.lang.Object
|-- MySQLInit
    |-- MySQLGetters
    |   |-- MySQLSetters
    |       |-- MySQLUpdater
    |           |-- MySQLDelete
    |               |-- com.ericsson.eos.database.MySQLConnector

All Implemented Interfaces:
    DBI, java.io.Serializable

< Constructors >

public class MySQLConnector
extends MySQLDelete
implements DBI, java.io.Serializable

The Class MySQLConnector.
Constructors

MySQLConnector

public MySQLConnector()

Instantiates a new MySQL connector (used for xmlParser connection).

MySQLConnector

public MySQLConnector(Model m)

Instantiates a new MySQL connector.

Parameters:

m - the model

com.ericsson.eos.database

Class MySQLDelete

java.lang.Object

|-- MySQLInit

|-- MySQLGetters

|-- MySQLSetters

|-- MySQLUpdater

|-- com.ericsson.eos.database.MySQLDelete

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

MySQLConnector

The Class MySQLDelete. This class handles all the delete functionality that the database supports. All the SQL-code needed to remove objects from the database is found here. This is part of the MySQL-inheritance chain.
MySQLDelete

public MySQLDelete()

Methods

deleteAlarm

public void deleteAlarm(java.lang.String notificationId)

Deletes the alarm with the given notificationId.

Parameters:
notificationId - the notification id

deleteCTP

public void deleteCTP(java.lang.String ems,
java.lang.String me,
java.lang.String ptp,
java.lang.String ctp)

Delete ctp. Deletes the CTP that is specified.

Parameters:
ems - the name of the EMS
me - the name of the ME
ptp - the name of the PTP/FTP
ctp - the name of the CTP that should be deleted

deleteEms

public void deleteEms(java.lang.String ems)

Delete ems. If the EMS is deleted all objects referring to the EMS are also deleted. This function should be used carefully.

Parameters:
ems - the name of the EMS
deleteME

public void deleteME(java.lang.String ems, java.lang.String me)

Delete me. Deletes the ManagedElement.

Parameters:
  
  ems - the name of the EMS
  me - the name of the ME that should be deleted

deleteMLSN

public void deleteMLSN(java.lang.String ems, java.lang.String mlsn)

Delete mlsn. This function deletes an MultiLayerSubnetwork, and all the elements under it.

Parameters:
  
  ems - the name of the EMS
  mlsn - the name of the subnet that should be deleted

deletePTP

public void deletePTP(java.lang.String ems, java.lang.String me, java.lang.String ptp)

Delete ptp. Deletes the PTP/FTP that is specified.

Parameters:
  
  ems - the name of the EMS
  me - the name of the ME
  ptp - the name and type of the PTP/FTP that should be deleted

deleteTCA

public void deleteTCA(java.lang.String notificationId)

Delete tca. Deletes the TCA with the given notificationId.

Parameters:
  
  notificationId - the notification id
deleteTL

public void deleteTL(java.lang.String ems,
                      java.lang.String tlName)

Delete tl. Deletes the Topological Link that is specified.

Parameters:
- ems - the name of the EMS
- tlName - the name of the TL that should be deleted.

com.ericsson.eos.database

Class MySQLGetters

java.lang.Object
   |-- MySQLInit
   |    |-- com.ericsson.eos.database.MySQLGetters

All Implemented Interfaces:
- java.io.Serializable

Direct Known Subclasses:
- MySQLSetters

< Constructors > < Methods >

public class MySQLGetters extends MySQLInit
                        implements java.io.Serializable

The Class MySQLGetters.

Constructors

MySQLGetters

public MySQLGetters()
**getActiveAlarms**

```java
public java.util.ArrayList getActiveAlarms(java.lang.String ems,
java.lang.String managedElement)
```

Returns all active alarms on the specified managedElement.

**Parameters:**
- ems - the ems
- managedElement - the managed element

**Returns:**
- the active alarms

---

**getActiveTCAs**

```java
public java.util.ArrayList getActiveTCAs(java.lang.String ems,
java.lang.String managedElement)
```

Returns all active TCAs given the specified managedElement.

**Parameters:**
- ems - the ems
- managedElement - the managed element

**Returns:**
- the active tc as

---

**getAdditionalInfo**

```java
public java.util.ArrayList getAdditionalInfo(int id)
```

Gets the additional info belonging to a given object name ID.

**Parameters:**
- id - the id

**Returns:**
- the additional info

---

**getAlarm**

```java
public java.util.HashMap getAlarm(java.lang.String notificationID)
```

Gets the alarm with the specified notificationID.

**Parameters:**
- notificationID - the notification id

**Returns:**
- the alarm
getAllActiveAlarms

public java.util.ArrayList getAllActiveAlarms()

Returns all NT_Alarms.

Returns:
the all active alarms

getAllActiveAlarmsFiltered

public java.util.ArrayList getAllActiveAlarmsFiltered(java.lang.String[] severityFilter,
java.lang.String[] probCauseFilter)

Returns alarms that don't contain severity values from severityFilter and probable causes from probCauseFilter.

Parameters:
severityFilter - the severity filter
probCauseFilter - the prob cause filter

Returns:
alarms filtered on severity and probable cause

getAllActiveTCAs

public java.util.ArrayList getAllActiveTCAs()

Returns all NT_TCA.

Returns:
the all active TCAs

getAllActiveTCAsFiltered

public java.util.ArrayList getAllActiveTCAsFiltered(java.lang.String[] severityFilter,
java.lang.String[] probCauseFilter)

Returns TCAs that don't contain severity values from severityFilter and probable causes from probCauseFilter.

Parameters:
severityFilter - the severity filter
probCauseFilter - the prob cause filter

Returns:
alarms filtered on severity and probable cause
getAllAlarmParameterNames

public java.util.ArrayList getAllAlarmParameterNames()

get all attribute-names associated with nt_alarm.

Returns:
the all alarm parameter names

getAllManagedElements

public java.util.ArrayList getAllManagedElements()

gets all managedElements.

Returns:
all managedElements

getAllManagedElements

public java.util.ArrayList getAllManagedElements(java.lang.String ems, java.lang.String subnetwork)

Gets all managed elements.

Parameters:
ems - the ems
subnetwork - the subnetwork

Returns:
the all managed elements

getAllTCAParameterNames

public java.util.ArrayList getAllTCAParameterNames()

gets all attribute-names associated with nt_tca.

Returns:
the all tca parameter names

getAllTopLevelSubnetworks

public java.util.ArrayList getAllTopLevelSubnetworks()

Returns information on all TopLevelSubnetworks.

Returns:
ArrayList a list of hashmaps containing all info on topLevelSubnetworks.
**getAllTopologicalLinks**

*public java.util.ArrayList getAllTopologicalLinks()*

returns all topologicalLinks.

**Returns:**

all topologicalLinks

---

**getCTP**


Gets the object structure given the CTP's objectName.

**Parameters:**

- ems - the ems
- me - the me
- ptp - the ptp
- ctp - the ctp

**Returns:**

the cTP

---

**getContainedCurrentTPNames**

*public java.lang.String[] getContainedCurrentTPNames(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)*

Retrieves the names of the Contained Current TPs filtered on the listed layer-rates. A current CTP is defined as either cross-connectable or cross-connected, in the current mapping configuration which can be seen in the attribute tpMappingmode (set to TM_NEITHER_TERMINATED_NORAVAILABLE_FOR_MAPPING or TM_NA). This means all potential CTPs except those that are Terminated and mapped.

**Parameters:**

- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**

the contained current tp names
getContainedCurrentTPs

```java
public java.util.ArrayList getContainedCurrentTPs(java.lang.String ems,
                                                java.lang.String me,
                                                java.lang.String tp,
                                                short[] layerRate)
```

Gets the object structures of all contained current CTPs filtered on the listed layer-rates. A current CTP is defined as either cross-connectable or cross-connected, in the current mapping configuration which can be seen in the attribute tpMappingmode (set to TM_NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING or TM_NA). This means all potential CTPs except those that are Terminated and mapped.

**Parameters:**
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**
- the contained current tps

getContainedInUseTPNames

```java
public java.lang.String[] getContainedInUseTPNames(java.lang.String ems,
                                                     java.lang.String me,
                                                     java.lang.String tp,
                                                     short[] layerRate)
```

Gets the names of contained In-Use CTPs filtered on the listed layer-rates. An In-Use CTP is defined as a CTP used in SNC (in any state) or a CTP that is terminated and mapped. These are CTPs used in SNCs as well as well as the CTPs with attribute tpMappingmode = TM_TERMINATED_AND_AVAILABLE_FOR_MAPPING.

**Parameters:**
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**
- the contained in use tp names
getContainedInUseTPs

public java.util.ArrayList getContainedInUseTPs(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the object structures of contained In-Use CTPs filtered on the listed layer-rates. An In-Use CTP is defined as a CTP used in SNC (in any state) or a CTP that is terminated and mapped. These are CTPs used in SNCs as well as well as the CTPs with attribute tpMappingmode = TM_TERMINATED_AND_AVAILABLE_FOR_MAPPING.

Parameters:
  ems - the ems
  me - the me
  tp - the tp
  layerRate - the layer rate

Returns:
  the contained in use tps

getContainedPotentialTPNames

public java.lang.String[] getContainedPotentialTPNames(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the name of all potential CTPs given the specified layerRates and level3Object.

Parameters:
  ems - the ems
  me - the me
  tp - the tp
  layerRate - the layer rate

Returns:
  the contained potential tp names
getContainedPotentialTPs

public java.util.ArrayList getContainedPotentialTPs(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

gets all potential CTPs given the specified layerRates and level3Object.

Parameters:
ems - the ems
me - the me
tp - the tp
layerRate - the layer rate

Returns:
the contained potential t ps

getEMSName

public java.lang.String getEMSName()

returns the name of the EMS.

Returns:
the name of the EMS

getEms

public java.util.HashMap getEms()

Returns the object structure of the EMS.

Returns:
all attributes of EMS

getLayerParameters

public java.util.ArrayList getLayerParameters(int id)

Gets the layer parameters belonging to the object with the given objectname ID.

Parameters:
id - the id

Returns:
the layer parameters
**getLayerRate**

```java
public java.lang.String getLayerRate(short s)
```

Gets the layer rate name given a short value. Most name-short mappings can be found in the TMF814 documentation.

**Parameters:**
- `s` - the s

**Returns:**
- the layer rate

---

**getManagedElement**

```java
public java.util.HashMap getManagedElement(java.lang.String ems, java.lang.String name)
```

Returns managedElement object structure given the ems and managed element name.

**Parameters:**
- `ems` - the ems
- `name` - the ME name

**Returns:**
- managedElement data

---

**getObjectName**

```java
public java.util.HashMap getObjectName(int id)
```

Gets the object name given the ID.

**Parameters:**
- `id` - the id

**Returns:**
- the object name
**getObjectNameID**

```java
public int getObjectNameID(java.lang.String lev1,
                          java.lang.String lev2,
                          java.lang.String lev3,
                          java.lang.String lev4)
```

Gets the ID number associated with the name of the object. Structured as a hierarchy of four levels with type&%&name values.

**Parameters:**
- lev1 - the lev1
- lev2 - the lev2
- lev3 - the lev3
- lev4 - the lev4

**Returns:**
- the object name id

**getPTP**

```java
public java.util.HashMap getPTP(java.lang.String ems,
                                 java.lang.String me,
                                 java.lang.String ptp)
```

Gets the PTP object structure given it's object name.

**Parameters:**
- ems - the ems
- me - the me
- ptp - the ptp

**Returns:**
- the pTP

**getPTPNames**

```java
public java.lang.String[] getPTPNames(java.lang.String ems,
                                       java.lang.String me,
                                       java.lang.String ptp)
```

return all level3object names (ptp,ftp) given ems and managedElement.

**Parameters:**
- ems - the ems
- me - the me

**Returns:**
- the pTP names
getPTPs

```java
public java.util.ArrayList getPTPs(java.lang.String ems,
    java.lang.String me)
```

gets all the PTPs/FTPs object structures under a managed element.

**Parameters:**
- ems - EMS that the system simulates.
- me - The specific managed element.

**Returns:**
All the ptps in hashmap for wrapped in a arraylist.

---

getSubNodes

```java
public java.util.ArrayList getSubNodes(java.lang.String subnet)
```

Returns all the managedElement names under a given subnetwork.

**Parameters:**
- subnet - the subnet

**Returns:**
the subnode names (managedElements) of an EMS

---

getSubnetwork

```java
public java.util.HashMap getSubnetwork(java.lang.String ems,
    java.lang.String mlsn)
```

Gets the subnetwork object structure given the object name.

**Parameters:**
- ems - the ems
- mlsn - the mlsn

**Returns:**
the subnetwork

---

getSupportedRates

```java
public java.util.HashMap getSupportedRates(int id)
```

Gets the supported rates belonging to the specified object name ID.

**Parameters:**
- id - the id

**Returns:**
the supported rates
getTCA

public java.util.HashMap getTCA(java.lang.String notificationID)

   Gets the TCA.

   Parameters:
   
   notificationID - the notification id

   Returns:
   
   the TCA

getTopologicalLink

public java.util.HashMap getTopologicalLink(int id)

   Gets the topological link object structure given the objectName ID.

   Parameters:
   
   id - the id

   Returns:
   
   the topological link

g getX733AdditionalInfo

public java.util.ArrayList getX733AdditionalInfo(java.lang.String id)

   Gets the x733 additional info given a notificationID.

   Parameters:
   
   id - the id

   Returns:
   
   the x733 additional info

g getX733MonitoredAttribute

public java.util.ArrayList getX733MonitoredAttribute(java.lang.String id)

   Gets the x733 monitored attributes given a notificationID.

   Parameters:
   
   id - the id

   Returns:
   
   the x733 monitored attribute
Class MySQLInit

java.lang.Object
    +--com.ericsson.eos.database.MySQLInit

All Implemented Interfaces:
    java.io.Serializable

Direct Known Subclasses:
    MySQLGetters

Constructors

MySQLInit

public MySQLInit()

Methods

connect

public void connect()

    Connects to the database using given username and password when interface was created.

disconnect

public void disconnect()

    Disconnects from the database. Use connect() to get a connection again.
**isActive**

```java
public boolean isActive()
```

Returns true if the connection is active.

**Returns:**
true, if is active

---

**reset**

```java
public void reset(java.lang.String fileName)
```

Reset the database. This function clears all the data from the database, use carefully. The file that is given should contain all the needed information to clear and rebuild the the structure again. In the case of this simulator a file containing the structure that first erases everything is used followed by a file for all the static data.

**Parameters:**
fileName - the file name of the sql file.

## com.ericsson.eos.database

### Class MySQLSetters

```java
java.lang.Object
   +--MySQLInit
   |   +--MySQLGetters
   |   +--com.ericsson.eos.database.MySQLSetters
```

**All Implemented Interfaces:**
java.io.Serializable

**Direct Known Subclasses:**
MySQLUpdater

< Constructors > < Methods >

**public class MySQLSetters**
extends MySQLGetters
implements java.io.Serializable

The Class MySQLSetters contains all the JDBC set-functions. It is part of an inheritance chain to MySQLConnector.
MySQLSetters

public MySQLSetters()

Methods

setAdditionalInfo

public boolean setAdditionalInfo(int objectNameID,
                                 java.lang.String[] addInfo)

Sets initial Additional Information Used when creating a new object.

Parameters:
    objectNameID - the object name id
    addInfo - the add info

Returns:
    true, if successful
public java.lang.String setCTP(java.lang.String emsName,
java.lang.String meName,
java.lang.String ptpName,
java.lang.String ctpName,
java.lang.String userLabel,
java.lang.String nativeEMSName,
java.lang.String owner,
java.lang.String type,
java.lang.String connectionState,
java.lang.String tpMappingMode,
java.lang.String direction,
java.lang.String tpProtectionAssociation,
java.lang.String edgePoint,
java.lang.String ingressName,
java.lang.String egressName,
java.lang.String[] addInfo)

Creates a new CTP.

Parameters:
emsName - the ems name
meName - the me name
ptpName - the ptp name
ctpName - the ctp name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
type - the type
connectionState - the connection state
tpMappingMode - the tp mapping mode
direction - the direction
tpProtectionAssociation - the tp protection association
edgePoint - the edge point
ingressName - the ingress name
egressName - the egress name
addInfo - the add info

Returns:
the string
setCrossConnection


Creates a CrossConnection.

Parameters:
- active - the active
- direction - the direction
- ccType - the cc type
- node1Ems - the node1 ems
- node1Me - the node1 me
- node1lev3 - the node1lev3
- node1lev4 - the node1lev4
- node2Ems - the node2 ems
- node2Me - the node2 me
- node2lev3 - the node2lev3
- node2lev4 - the node2lev4
- additionalInfo - the additional info

Returns:
- the string

setEMS


Creates a new EMS.

Parameters:
- emsName - the ems name
- userLabel - the user label
- nativeEMSName - the native ems name
- owner - the owner
- emsVersion - the ems version
- type - the type
- additionalInfo - ("name" + &%& +"val")

Returns:
- the string
setLayerParameters

public java.lang.String setLayerParameters(java.lang.String lev1,
java.lang.String lev2,
java.lang.String lev3,
java.lang.String lev4,
java.lang.String layer,
java.lang.String[] paramName,
java.lang.String[] paramVal)

Sets the layer parameters.

Parameters:
lev1 - the lev1
lev2 - the lev2
lev3 - the lev3
lev4 - the lev4
layer - the layer
paramName - the param name
paramVal - the param val

Returns:
the string
setManagedElement

public java.lang.String setManagedElement(java.lang.String emsName,
                                             java.lang.String meName,
                                             java.lang.String subnetName,
                                             java.lang.String userLabel,
                                             java.lang.String location,
                                             java.lang.String version,
                                             java.lang.String productName,
                                             java.lang.String communicationState,
                                             java.lang.String nativeEMSName,
                                             java.lang.String emsInSyncState,
                                             java.lang.String owner,
                                             java.lang.String[] supportedRates,
                                             java.lang.String[] additionalInfo)

Creates a new ManagedElement.

Parameters:
   emsName - the ems name
   meName - the me name
   subnetName - the subnet name
   userLabel - the user label
   location - the location
   version - the version
   productName - the product name
   communicationState - the communication state
   nativeEMSName - the native ems name
   emsInSyncState - the ems in sync state
   owner - the owner
   supportedRates - the supported rates
   additionalInfo - the additional info

Returns:
   the string
setNT_Alarms

public java.lang.String setNT_Alarms(java.lang.String notificationId,
java.lang.String emsName,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String layerRate,
java.lang.String perceivedSeverity,
java.lang.String acknowledgeIndication,
java.lang.String nativeProbableCause,
java.lang.String probableCause,
java.lang.String probableCauseQualifier,
java.lang.String serviceAffecting,
java.lang.String additionalText,
java.lang.String rcaiIndicator,
java.lang.String[] x733AdditionalInfo,
java.lang.String x733BackedUpStatus,
java.lang.String x733BackupObject,
java.lang.String[] x733CorrelatedNotifications,
java.lang.String x733EventTypes,
java.lang.String[] x733MonitoredAttributes,
java.lang.String[] x733ProposedRepairActions,
java.lang.String[] x733SpecificProblems,
java.lang.String x733TrendIndications,
java.lang.String[] affectedTPList)

Sets alarms of type NT_ALARM. Use "" for a value not used, do not use NULL.

Parameters:
  notificationId - the notification id
  emsName - the ems name
  level2Object - the level2 object
  level3Object - the level3 object
  level4Object - the level4 object
  nativeEmsName - the native ems name
  objectType - the object type
  objectTypeQualifier - the object type qualifier
  emsTime - the ems time
  neTime - the ne time
  isClearable - the is clearable
  layerRate - the layer rate
  perceivedSeverity - the perceived severity
  acknowledgeIndication - the acknowledge indication
  nativeProbableCause - the native probable cause
  probableCause - the probable cause
  probableCauseQualifier - the probable cause qualifier
  serviceAffecting - the service affecting
  additionalText - the additional text
  rcaiIndicator - the rcai indicator
  x733AdditionalInfo - the x733 additional info
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733CorrelatedNotifications - the x733 correlated notifications
x733EventType - the x733 event type
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733SpecificProblems - the x733 specific problems
x733TrendIndication - the x733 trend indication
affectedTPList - the affected tp list

Returns:

the string
setNT_TCA

public java.lang.String setNT_TCA(java.lang.String notificationId,
java.lang.String emsName,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String perceivedSeverity,
java.lang.String layerRate,
java.lang.String granularity,
java.lang.String pmParameterName,
java.lang.String pmLocation,
java.lang.String thresholdType,
java.lang.String floatValue,
java.lang.String unit,
java.lang.String acknowledgeIndication)

Creates alarm of type NT_TCA.

Parameters:

- notificationId - the notification id
- emsName - the ems name
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object
- nativeEmsName - the native ems name
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- perceivedSeverity - the perceived severity
- layerRate - the layer rate
- granularity - the granularity
- pmParameterName - the pm parameter name
- pmLocation - the pm location
- thresholdType - the threshold type
- floatValue - the float value
- unit - the unit
- acknowledgeIndication - the acknowledge indication

Returns:

the string
public boolean setObjectName(java.lang.String lev1,
                     java.lang.String lev2,
                     java.lang.String lev3,
                     java.lang.String lev4)

Creates a new objectName for any type of object. level1 contains the name of the EMS, level2 contains the name of the MultiLayerSubnetwork, ManagedElement or TopologicalLink, level3 contains the name of the PTP/FTP or CrossConnection and level 4 contains the name of CTP.

Parameters:
lev1 - the lev1
lev2 - the lev2
lev3 - the lev3
lev4 - the lev4

Returns:
true, if successful
setPTP

public java.lang.String setPTP(java.lang.String emsName,
java.lang.String meName,
java.lang.String ptpName,
java.lang.String userLabel,
java.lang.String nativeEMSName,
java.lang.String owner,
java.lang.String type,
java.lang.String connectionState,
java.lang.String tpMappingMode,
java.lang.String direction,
java.lang.String tpProtectionAssociation,
java.lang.String edgePoint,
java.lang.String ingressName,
java.lang.String egressName,
java.lang.String[] additionalInfo)

Creates a new PTP/FTP.

Parameters:

emsName - the ems name
meName - the me name
ptpName - the ptp name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
type - the type
connectionState - the connection state
tpMappingMode - the tp mapping mode
direction - the direction
tpProtectionAssociation - the tp protection association
direction - the direction
ingressName - the ingress name
egressName - the egress name
additionalInfo - the additional info

Returns:
the string

setSupportedRates

public boolean setSupportedRates(int objectNameID,
java.lang.String[] supportedRates)

Sets the supported rates for a given ObjectName ID.

Parameters:

objectNameID - the object name id
supportedRates - the supported rates

Returns:
true, if successful

Creates a new top level subnetwork.

Parameters:
- emsName - the ems name
- subnetName - the subnet name
- owner - the owner
- userLabel - the user label
- nativeEMSName - the native ems name
- subnetworkType - the subnetwork type
- supportedRates - the supported rates
- additionalInfo - the additional info

Returns:
the string
**setTopologicalLink**

```java
public java.lang.String setTopologicalLink(java.lang.String emsName,
                                           java.lang.String tlName,
                                           java.lang.String userLabel,
                                           java.lang.String nativeEmsName,
                                           java.lang.String owner,
                                           java.lang.String direction,
                                           java.lang.String rate,
                                           java.lang.String node1Ems,
                                           java.lang.String node1Me,
                                           java.lang.String node1PTP,
                                           java.lang.String node1ctp,
                                           java.lang.String node2Ems,
                                           java.lang.String node2Me,
                                           java.lang.String node2PTP,
                                           java.lang.String node2ctp,
                                           java.lang.String[] additionalInfo)
```

creates a new TopologicalLink.

**Parameters:**
- emsName - the ems name
- tlName - the tl name
- userLabel - the user label
- nativeEmsName - the native ems name
- owner - the owner
- direction - the direction
- rate - the rate
- node1Ems - the node1 ems
- node1Me - the node1 me
- node1PTP - the node1 ptp
- node1ctp - the node1ctp
- node2Ems - the node2 ems
- node2Me - the node2 me
- node2PTP - the node2 ptp
- node2ctp - the node2ctp
- additionalInfo - the additional info

**Returns:**
- the string

---

**setX733AdditionalInfo**

```java
public boolean setX733AdditionalInfo(java.lang.String notificationID,
                                       java.lang.String[] x733AdditionalInfo)
```

Sets the x733 additional info.

**Parameters:**
- notificationID - the notification id
- x733AdditionalInfo - the x733 additional info

**Returns:**
- true, if successful
**setX733MonitoredAttribute**

```java
public boolean setX733MonitoredAttribute(java.lang.String notificationID,
java.lang.String[] x733MonitoredAttributes)
```

Sets the x733 monitored attribute.

**Parameters:**
- notificationID - the notification id
- x733MonitoredAttributes - the x733 monitored attributes

**Returns:**
- true, if successful

---

**com.ericsson.eos.database**

**Class MySQLUpdater**

java.lang.Object

|-- MySQLInit

|-- MySQLGetters

|-- MySQLSetters

|-- com.ericsson.eos.database.MySQLUpdater

**All Implemented Interfaces:**
- java.io.Serializable

**Direct Known Subclasses:**
- MySQLDelete

< Constructors > < Methods >

**public class MySQLUpdater**
extends MySQLSetters
implements java.io.Serializable

The Class MySQLUpdater. This class contains all the functions for updating data already in the database.

**Constructors**

**MySQLUpdater**

```java
public MySQLUpdater()
```

**Methods**
concat

public java.lang.String concat(java.lang.String[] list)

Concat.

Parameters:
  list - the list

Returns:
  the string

updateCTP

public void updateCTP(java.lang.String emsName,
                      java.lang.String meName,
                      java.lang.String ptpName,
                      java.lang.String ctpName,
                      java.lang.String newCTPName,
                      java.lang.String userLabel,
                      java.lang.String nativeEMSName,
                      java.lang.String owner,
                      java.lang.String type,
                      java.lang.String connectionState,
                      java.lang.String tpMappingMode,
                      java.lang.String direction,
                      java.lang.String tpProtectionAssociation,
                      java.lang.String edgePoint,
                      java.lang.String ingressName,
                      java.lang.String egressName,
                      java.lang.String[] transmissionParameters,
                      java.lang.String[] additionalInfo)

Updatectp.

Parameters:
  emsName - the ems name
  meName - the me name
  ptpName - the ptp name
  ctpName - the ctp name
  newCTPName - the new ctp name
  userLabel - the user label
  nativeEMSName - the native ems name
  owner - the owner
  type - the type
  connectionState - the connection state
  tpMappingMode - the tp mapping mode
  direction - the direction
  tpProtectionAssociation - the tp protection association
  edgePoint - the edge point
  ingressName - the ingress name
  egressName - the egress name
  transmissionParameters - the transmission parameters
  additionalInfo - the additional info
updateEMS

public void updateEMS(java.lang.String emsName,
          java.lang.String newEmsName,
          java.lang.String userLabel,
          java.lang.String nativeEMSName,
          java.lang.String owner,
          java.lang.String emsVersion,
          java.lang.String type,
          java.lang.String[] additionalInfo)

Update ems.

Parameters:

emsName - the ems name
newEmsName - the new ems name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
emsVersion - the ems version
type - the type
additionalInfo - the additional info

updateManagedElement

public void updateManagedElement(java.lang.String emsName,
          java.lang.String meName,
          java.lang.String newMeName,
          java.lang.String subnetName,
          java.lang.String userLabel,
          java.lang.String location,
          java.lang.String version,
          java.lang.String productName,
          java.lang.String communicationState,
          java.lang.String nativeEMSName,
          java.lang.String emsInSyncState,
          java.lang.String owner,
          java.lang.String[] supportedRates,
          java.lang.String[] additionalInfo)

Update managed element.

Parameters:

emsName - the ems name
meName - the me name
newMeName - the new me name
subnetName - the subnet name
userLabel - the user label
location - the location
version - the version
productName - the product name
communicationState - the communication state
nativeEMSName - the native ems name
emsInSyncState - the ems in sync state
owner - the owner
supportedRates - the supported rates
additionalInfo - the additional info

Update multi layer subnetwork.

**Parameters:**
- level1Object - the level1 object
- level2Object - the level2 object
- newLevel2Object - the new level2 object
- userLabel - the user label
- owner - the owner
- nativeEmsName - the native ems name
- subnetworkType - the subnetwork type
- layerRate - the layer rate
- additionalInfo - the additional info
updateNTAlarm

public java.lang.String updateNTAlarm(java.lang.String notificationId,
        java.lang.String ems,
        java.lang.String level2Object,
        java.lang.String level3Object,
        java.lang.String level4Object,
        java.lang.String nativeEmsName,
        java.lang.String objectType,
        java.lang.String objectTypeQualifier,
        java.lang.String emsTime,
        java.lang.String neTime,
        java.lang.String isClearable,
        java.lang.String layerRate,
        java.lang.String perceivedSeverity,
        java.lang.String acknowledgeIndication,
        java.lang.String nativeProbableCause,
        java.lang.String probableCause,
        java.lang.String probableCauseQualifier,
        java.lang.String serviceAffecting,
        java.lang.String additionalText,
        java.lang.String rcaiIndicator,
        java.lang.String[] x733AdditionalInfo,
        java.lang.String x733BackedUpStatus,
        java.lang.String x733BackupObject,
        java.lang.String[] x733CorrelatedNotifications,
        java.lang.String x733EventType,
        java.lang.String[] x733MonitoredAttributes,
        java.lang.String[] x733ProposedRepairActions,
        java.lang.String[] x733SpecificProblems,
        java.lang.String x733TrendIndication,
        java.lang.String[] affectedTPList)

Update nt alarm.

Parameters:

notificationId - the notification id
ems - the ems
level2Object - the level2 object
level3Object - the level3 object
level4Object - the level4 object
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
layerRate - the layer rate
perceivedSeverity - the perceived severity
acknowledgeIndication - the acknowledge indication
nativeProbableCause - the native probable cause
probableCause - the probable cause
probableCauseQualifier - the probable cause qualifier
serviceAffecting - the service affecting
additionalText - the additional text
rcaiIndicator - the rcai indicator
x733AdditionalInfo - the x733 additional info
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733CorrelatedNotifications - the x733 correlated notifications
x733EventType - the x733 event type
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733SpecificProblems - the x733 specific problems
x733TrendIndication - the x733 trend indication
affectedTPList - the affected tp list

Returns:

the string
updateNTTCA

public java.lang.String updateNTTCA(java.lang.String notificationId,
java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String perceivedSeverity,
java.lang.String layerRate,
java.lang.String granularity,
java.lang.String pmParameterName,
java.lang.String pmLocation,
java.lang.String thresholdType,
java.lang.String value,
java.lang.String unit,
java.lang.String acknowledgeIndication)

Update nttca.

Parameters:

- notificationId - the notification id
- ems - the ems
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object
- nativeEmsName - the native ems name
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- perceivedSeverity - the perceived severity
- layerRate - the layer rate
- granularity - the granularity
- pmParameterName - the pm parameter name
- pmLocation - the pm location
- thresholdType - the threshold type
- value - the value
- unit - the unit
- acknowledgeIndication - the acknowledge indication

Returns:

the string
public void updatePTP(java.lang.String emsName,
    java.lang.String meName,
    java.lang.String ptpName,
    java.lang.String newPTPName,
    java.lang.String userLabel,
    java.lang.String nativeEMSName,
    java.lang.String owner,
    java.lang.String type,
    java.lang.String connectionState,
    java.lang.String tpMappingMode,
    java.lang.String direction,
    java.lang.String tpProtectionAssociation,
    java.lang.String edgePoint,
    java.lang.String ingressName,
    java.lang.String egressName,
    java.lang.String[] transmissionParameters,
    java.lang.String[] additionalInfo)

Update ptp.

**Parameters:**

- emsName - the ems name
- meName - the me name
- ptpName - the ptp name
- newPTPName - the new ptp name
- userLabel - the user label
- nativeEMSName - the native ems name
- owner - the owner
- type - the type
- connectionState - the connection state
- tpMappingMode - the tp mapping mode
- direction - the direction
- tpProtectionAssociation - the tp protection association
- edgePoint - the edge point
- ingressName - the ingress name
- egressName - the egress name
- transmissionParameters - the transmission parameters
- additionalInfo - the additional info
public void updateTopologicalLink(java.lang.String emsName,
java.lang.String tlName,
java.lang.String newTlName,
java.lang.String userLabel,
java.lang.String nativeEmsName,
java.lang.String owner,
java.lang.String direction,
java.lang.String rate,
java.lang.String node1Ems,
java.lang.String node1Me,
java.lang.String node1ptp,
java.lang.String node1ctp,
java.lang.String node2Ems,
java.lang.String node2Me,
java.lang.String node2ptp,
java.lang.String node2ctp,
java.lang.String[] additionalInfo)

Update topological link.

Parameters:

emsName - the ems name
tlName - the tl name
newTlName - the new tl name
userLabel - the user label
nativeEmsName - the native ems name
owner - the owner
direction - the direction
rate - the rate
node1Ems - the node1 ems
node1Me - the node1 me
node1ptp - the node1ptp
node1ctp - the node1ctp
node2Ems - the node2 ems
node2Me - the node2 me
node2ptp - the node2ptp
node2ctp - the node2ctp
additionalInfo - the additional info
public class XmlParser
extends java.util.Observable
implements java.io.Serializable, java.langRunnable

The Class XmlParser.

Fields

cancel

public boolean cancel
    The cancel.

Constructors

XmlParser

public XmlParser()
    Instantiates a new xml parser. This is used from the command-line.

XmlParser

public XmlParser(java.lang.String path)
    Instantiates a new xml parser given the path. This is used from the GUI.
    Parameters:
        path - the path

Methods

clearDB

public void clearDB()
    Clear db.
countFiles

public int countFiles(java.lang.String type)

    Count files.

    Parameters:
    type - the type

    Returns:
    the int

main

public static void main(java.lang.String[] args)

    The main method.

    Parameters:
    args - the arguments

run

public void run()
Package com.ericsson.eos.debugger

Class Summary

**Debugger**

Debugger is a class developed for testing purpose only.

com.ericsson.eos.debugger

Class Debugger

```
java.lang.Object
   |-- java.util.Observable
   |   |-- com.ericsson.eos.debugger.Debugger
```

All Implemented Interfaces:

- DebugInterface
- java.io.Serializable

< Constructors > < Methods >

public class Debugger
extends java.util.Observable
implements DebugInterface, java.io.Serializable

Debugger is a class developed for testing purpose only. This class can either print event messages to file, screen(command-line), both or not print at all.

Constructors

**Debugger**

```java
public Debugger(Config conf)
```

Instantiates a new debugger.

**Parameters:**

- conf - the conf

Methods

**changedConf**

```java
public void changedConf(Config conf)
```

Changed conf.

**Parameters:**

- conf - the conf
**getMessages**

public java.lang.String[] getMessages()

Gets the messages.

**Returns:**

the messages

---

**setDebugToFile**

public void setDebugToFile(boolean bool)

setDebugToFile turns on the feature to store the different events that occurs to a file. The file can be located at debugEOS.log.

**Parameters:**

bool - the new debug to file

---

**setDebugToScreen**

public void setDebugToScreen(boolean bool)

setDebugToScreen turns on the feature to print the different events that occurs to the standard output. This is usually the command-line.

**Parameters:**

bool - the new debug to screen

---

**setMessages**

public void setMessages(java.lang.String[] messages)

Sets the messages.

**Parameters:**

messages - the new messages
write

```java
public void write(java.lang.String str,
                  java.lang.Object o,
                  java.util.logging.Level level)
```

write handles all the incoming messages and events. This is the function that all "debugging" events should use. By giving this function a good description of the event that occurs debugging will hopefully be a lot easier. Time-stamp is automatically added.

**Parameters:**

- `str` - The string that contains the message or describes the event that should be logged/printed.
- `o` - The object or class that logged the message. This is where the message originates from.
- `level` - the level
Package com.ericsson.eos.dynamicLoader

Class Summary
DynamicLoader

Handles the loading of input modules dynamically.

com.ericsson.eos.dynamicLoader

Class DynamicLoader

java.lang.Object

+--com.ericsson.eos.dynamicLoader.DynamicLoader

< Constructors > < Methods >

public class DynamicLoader
extends java.lang.Object

Handles the loading of input modules dynamically.

Author:
emikrie, Mikael Riedel

Constructors

DynamicLoader

public DynamicLoader(Model m)

Instantiates a new dynamic loader.

Parameters:

m - the model

Methods
loadModules

public java.util.ArrayList loadModules()

Load modules. This function loads all the modules that are inside a Jar-file and that complies with the NBI-interface from the folder NBI.

Returns:

the array list
Package com.ericsson.eos.helper

Class Summary

InputChecker
The Class InputChecker.

Splitter
The Class Splitter.

TreePathDivider
The Class TreePathDivider.

com.ericsson.eos.helper

Class InputChecker

java.lang.Object
   +--com.ericsson.eos.helper.InputChecker

< Methods >

public class InputChecker
extends java.lang.Object

The Class InputChecker.

Methods

checkInteger

public static boolean checkInteger(java.lang.String str)

Check integer. Returns true if the string contains only a integer.

Parameters:
   str - the str

Returns:
   true, if successful
checkShort

public static boolean checkShort(java.lang.String str)

Check short. Returns true if the string contains only a short.

Parameters:

str - the str

Returns:

true, if successful

com.ericsson.eos.helper

Class Splitter

java.lang.Object

|--com.ericsson.eos.helper.Splitter

< Methods >

public class Splitter extends java.lang.Object

The Class Splitter.

Methods

extensionFinder

public static java.lang.String extensionFinder(java.lang.String filePath)

Extension finder. This takes a path and only returns the file Extension of the file the path is pointing at.

Parameters:

filePath - the file path

Returns:

the string
removeHardParentheses
public static java.lang.String removeHardParentheses(java.lang.String str)

Removes the hard parentheses.

Parameters:
str - the str

Returns:
the string

removeParentheses
public static java.lang.String removeParentheses(java.lang.String str)

Removes the parentheses.

Parameters:
str - the str

Returns:
the string

splitParentheses
public static java.lang.String[] splitParentheses(java.lang.String str)

Split parentheses. Removes all parentheses and removes everything else as a list of strings.

Parameters:
str - the str

Returns:
the string[]

com.ericsson.eos.helper

Class TreePathDivider

java.lang.Object
   +--com.ericsson.eos.helper.TreePathDivider

< Methods >

public class TreePathDivider
extends java.lang.Object

The Class TreePathDivider.
Methods

getDepth

public static int getDepth(java.lang.String str)

Gets the depth

Parameters:

str - the str

Returns:

the depth

g getLast

public static java.lang.String getLast(java.lang.String path)

Gets the last selected element

Parameters:

path - the path

Returns:

the last

g getList

public static java.lang.String[][] getList(java.lang.String str)

Gets the list. returns a list of the tree path.

Parameters:

str - the str

Returns:

the list
Package com.ericsson.eos.interfaces

Interface Summary

**DBI**
The Interface DBI.

**DebugInterface**
The Interface DebugInterface.

**NBI**
This interface should be implemented by all classes that want to use the simulator for north-bound communication.

---

com.ericsson.eos.interfaces

**Interface DBI**

< Methods >

```java
public interface DBI
```
The Interface DBI.

---

**Methods**

**connect**
```java
public void connect()
```
Connects to the database using given username and password when interface was created.

**deleteAlarm**
```java
public void deleteAlarm(java.lang.String notificationId)
```
Delete alarm.

**Parameters:**
- `notificationId` - the notification id
deleteCTP

public void deleteCTP(java.lang.String ems,
                        java.lang.String me,
                        java.lang.String ptp,
                        java.lang.String ctp)

Delete ctp.

**Parameters:**
- ems - the ems
- me - the me
- ptp - the ptp
- ctp - the ctp

deleteEms

public void deleteEms(java.lang.String ems)

Delete ems.

**Parameters:**
- ems - the ems

deleteME

public void deleteME(java.lang.String ems,
                        java.lang.String me)

Delete me.

**Parameters:**
- ems - the ems
- me - the me

deleteMLSN

public void deleteMLSN(java.lang.String ems,
                        java.lang.String mlsn)

Delete mlsn.

**Parameters:**
- ems - the ems
- mlsn - the mlsn
**deletePTP**

```java
public void deletePTP(java.lang.String ems,
                      java.lang.String me,
                      java.lang.String ptp)
```

Delete ptp.

**Parameters:**
- ems - the ems
- me - the me
- ptp - the ptp

---

**deleteTCA**

```java
public void deleteTCA(java.lang.String notificationId)
```

Delete tca.

**Parameters:**
- notificationId - the notification id

---

**deleteTL**

```java
public void deleteTL(java.lang.String ems,
                     java.lang.String tlName)
```

Delete tl.

**Parameters:**
- ems - the ems
- tlName - the tl name

---

**disconnect**

```java
public void disconnect()
```

Disconnects from the database. Use connect() to get a connection again.
**getActiveAlarms**

```java
public java.util.ArrayList getActiveAlarms(java.lang.String ems,
                                           java.lang.String managedElement)
```

Retrieves all active EMS and ME active alarms on the specified managedElement.

**Parameters:**
- `ems` - the ems
- `managedElement` - the managed element

**Returns:**
- an ArrayList of Alarms, as HashMaps containing the attributes and values.

---

**getActiveTCAs**

```java
public java.util.ArrayList getActiveTCAs(java.lang.String ems,
                                        java.lang.String managedElement)
```

Gets the active tc as.

**Parameters:**
- `ems` - the ems
- `managedElement` - the managed element

**Returns:**
- the active tc as

---

**getAdditionalInfo**

```java
public java.util.ArrayList getAdditionalInfo(int id)
```

Gets the additional info.

**Parameters:**
- `id` - the id

**Returns:**
- the additional info

---

**getAlarm**

```java
public java.util.HashMap getAlarm(java.lang.String notificationID)
```

Retrieves alarm information of a specified alarm.

**Parameters:**
- `notificationID` - the notification id

**Returns:**
- a HashMap containing the attributes and values of the specified alarm.
getAllActiveAlarms

public java.util.ArrayList getAllActiveAlarms()

Gets the all active alarms.

Returns:
the all active alarms

getAllActiveAlarmsFiltered

public java.util.ArrayList getAllActiveAlarmsFiltered(java.lang.String[] severityFilter,
java.lang.String[] probCauseFilter)

Retrieves alarms that don’t contain severity values from severityFilter and probable causes from probCauseFilter.

Parameters:
severityFilter - the serverity filter
probCauseFilter - the prob cause filter

Returns:
the all active alarms filtered

ggetAllActiveTCAs

public java.util.ArrayList getAllActiveTCAs()

Gets the all active tcas.

Returns:
the all active tcas

ggetAllActiveTCAsFiltered

public java.util.ArrayList getAllActiveTCAsFiltered(java.lang.String[] severityFilter,
java.lang.String[] probCauseFilter)

Gets the all active tc as filtered.

Parameters:
severityFilter - the serverity filter
probCauseFilter - the prob cause filter

Returns:
the all active tc as filtered
**getAllAlarmParameterNames**

```java
public java.util.ArrayList getAllAlarmParameterNames()
```

Gets the all alarm parameter names.

**Returns:**
the all alarm parameter names

---

**getAllManagedElements**

```java
public java.util.ArrayList getAllManagedElements()
```

Retrieves the object structures of all managedElements.

**Returns:**
an ArrayList of Managed elements.

---

**getAllManagedElements**

```java
public java.util.ArrayList getAllManagedElements(java.lang.String ems, java.lang.String subnetwork)
```

Gets the all managed elements.

**Parameters:**
- ems - the ems
- subnetwork - the subnetwork

**Returns:**
the all managed elements

---

**getAllTCAParameterNames**

```java
public java.util.ArrayList getAllTCAParameterNames()
```

Gets the all tca parameter names.

**Returns:**
the all tca parameter names

---

**getAllTopLevelSubnetworks**

```java
public java.util.ArrayList getAllTopLevelSubnetworks()
```

Gets the all top level subnetworks.

**Returns:**
the all top level subnetworks
getAllTopologicalLinks

public java.util.ArrayList getAllTopologicalLinks()

Gets the all topological links.

Returns:
the all topological links

getCTP

public java.util.HashMap getCTP(java.lang.String ems,
java.lang.String me,
java.lang.String ptp,
java.lang.String ctp)

Gets the cTP.

Parameters:
ems - the ems
me - the me
ptp - the ptp
ctp - the ctp

Returns:
the cTP

getContainedCurrentTPNames

public java.lang.String[] getContainedCurrentTPNames(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

Gets the contained current tp names.

Parameters:
ems - the ems
me - the me
tp - the tp
layerRate - the layer rate

Returns:
the contained current tp names
### getContainedCurrentTPs

```java
public java.util.ArrayList getContainedCurrentTPs(java.lang.String ems,
                java.lang.String me,
                java.lang.String tp,
                short[] layerRate)
```

Gets the contained current tps.

**Parameters:**
- `ems` - the ems
- `me` - the me
- `tp` - the tp
- `layerRate` - the layer rate

**Returns:**
- the contained current tps

### getContainedInUseTPNames

```java
public java.lang.String[] getContainedInUseTPNames(java.lang.String ems,
                                                      java.lang.String me,
                                                      java.lang.String tp,
                                                      short[] layerRate)
```

Gets the contained in use tp names.

**Parameters:**
- `ems` - the ems
- `me` - the me
- `tp` - the tp
- `layerRate` - the layer rate

**Returns:**
- the contained in use tp names

### getContainedInUseTPs

```java
public java.util.ArrayList getContainedInUseTPs(java.lang.String ems,
                                                java.lang.String me,
                                                java.lang.String tp,
                                                short[] layerRate)
```

Gets the contained in use tps.

**Parameters:**
- `ems` - the ems
- `me` - the me
- `tp` - the tp
- `layerRate` - the layer rate

**Returns:**
- the contained in use tps
getContainedPotentialTPNames

public java.lang.String[] getContainedPotentialTPNames(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the contained potential tp names.

Parameters:

- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

Returns:

the contained potential tp names

getContainedPotentialTPs

public java.util.ArrayList getContainedPotentialTPs(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the contained potential tps.

Parameters:

- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

Returns:

the contained potential tps

getEMSName

public java.lang.String getEMSName()

Retrieves the EMS name.

Returns:

the EMS name.
getEms
public java.util.HashMap getEms()

Retrieves the EMS.

Returns:
    a Hashmap of attributes.

getLayerParameters
public java.util.ArrayList getLayerParameters(int id)

Gets the layer parameters.

Parameters:
    id - the id

Returns:
    the layer parameters

getLayerRate
public java.lang.String getLayerRate(short s)

Gets the layer rate.

Parameters:
    s - the s

Returns:
    the layer rate

getManagedElement
public java.util.HashMap getManagedElement(java.lang.String ems,
                                           java.lang.String name)

Retrieves the managedElement information for a given objectName.

Parameters:
    ems - the ems
    name - the name

Returns:
    a Hashmap of attributes.
**getObjectName**

```java
public java.util.HashMap getObjectName(int id)
```

Retrieves the objectName that belongs to the given id.

**Parameters:**
- id - the id

**Returns:**
a Hashmap containing the hierarchical structure of level and name values.

---

**getObjectNameID**

```java
public int getObjectNameID(java.lang.String level1Object,
                            java.lang.String level2Object,
                            java.lang.String level3Object,
                            java.lang.String level4Object)
```

Retrieves the ID of the objectname given the name in the form of a hierarchical structure.

**Parameters:**
- level1Object - the level1 object
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object

**Returns:**
the object name id

---

**getPTP**

```java
public java.util.HashMap getPTP(java.lang.String ems,
                                 java.lang.String me,
                                 java.lang.String ptp)
```

Gets the pTP.

**Parameters:**
- ems - the ems
- me - the me
- ptp - the ptp

**Returns:**
the pTP
getPTPNames

public java.lang.String[] getPTPNames(java.lang.String ems, java.lang.String me)

Gets the pTP names.

**Parameters:**
- ems - the ems
- me - the me

**Returns:**
the pTP names

getPTPs

public java.util.ArrayList getPTPs(java.lang.String ems, java.lang.String me)

Gets the pT ps.

**Parameters:**
- ems - the ems
- me - the me

**Returns:**
the pT ps

getSubNodes

public java.util.ArrayList getSubNodes(java.lang.String subnet)

Retrieves all ManagedElements under a given subnetwork.

**Parameters:**
- subnet - the subnet

**Returns:**
an ArrayList of Names.
getSubnetwork

public java.util.HashMap getSubnetwork(java.lang.String ems,
java.lang.String mlsn)

Gets the subnetwork.

Parameters:

- ems - the ems
- mlsn - the mlsn

Returns:

the subnetwork

getSupportedRates

public java.util.HashMap getSupportedRates(int id)

Gets the supported rates.

Parameters:

- id - the id

Returns:

the supported rates

getTCA

public java.util.HashMap getTCA(java.lang.String notificationID)

Gets the tCA.

Parameters:

- notificationID - the notification id

Returns:

the tCA

getTopologicalLink

public java.util.HashMap getTopologicalLink(int id)

Gets the topological link.

Parameters:

- id - the id

Returns:

the topological link
**getX733AdditionalInfo**

public java.util.ArrayList getX733AdditionalInfo(java.lang.String id)

  Gets the x733 additional info.

  **Parameters:**
  id - the id

  **Returns:**
  the x733 additional info

**getX733MonitoredAttribute**

public java.util.ArrayList getX733MonitoredAttribute(java.lang.String id)

  Gets the x733 monitored attribute.

  **Parameters:**
  id - the id

  **Returns:**
  the x733 monitored attribute

**isActive**

public boolean isActive()

  Checks if is active.

  **Returns:**
  true, if is active

**reset**

public void reset(java.lang.String fileName)

  Resets database to default structure with empty tables.

  **Parameters:**
  fileName - the file name
setCTP

public java.lang.String setCTP(java.lang.String ems,
        java.lang.String me,
        java.lang.String ptp,
        java.lang.String ctp,
        java.lang.String userLabel,
        java.lang.String nativeEMSName,
        java.lang.String owner,
        java.lang.String type,
        java.lang.String connectionState,
        java.lang.String tpMappingMode,
        java.lang.String direction,
        java.lang.String tpProtectionAssociation,
        java.lang.String edgePoint,
        java.lang.String ingressName,
        java.lang.String egressName,
        java.lang.String[] addInfo)

Creates a new CTP.

Parameters:

ems - the ems
me - the me
ptp - the ptp
ctp - the ctp
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
type - the type
connectionState - the connection state
tpMappingMode - the tp mapping mode
direction - the direction
tpProtectionAssociation - the tp protection association
depoint - the edge point
 ingressName - the ingress name
egressName - the egress name
addInfo - the add info

Returns:

either null or an error message
**setCrossConnection**

```java
```

Creates a new CrossConnection.

**Parameters:**
- active - the active
- direction - the direction
- ccType - the cc type
- node1Ems - the node1 ems
- node1Me - the node1 me
- node1lev3 - the node1lev3
- node1lev4 - the node1lev4
- node2Ems - the node2 ems
- node2Me - the node2 me
- node2lev3 - the node2lev3
- node2lev4 - the node2lev4
- addInfo - the add info

**Returns:**
- either null or an error message

---

**setEMS**

```java
```

Creates a new EMS.

**Parameters:**
- ems - the ems
- userLabel - the user label
- nativeEMSName - the native ems name
- owner - the owner
- emsVersion - the ems version
- type - the type
- addInfo - the add info

**Returns:**
- either null or an error message
**setLayerParameters**

```java
public java.lang.String setLayerParameters(java.lang.String ems,
                                          java.lang.String me,
                                          java.lang.String ptp,
                                          java.lang.String ctp,
                                          java.lang.String layer,
                                          java.lang.String[] paramName,
                                          java.lang.String[] paramVal)
```

For a specified TP, a certain layer can be given several parameter names and values.

**Parameters:**
- `ems` - the ems
- `me` - the me
- `ptp` - the ptp
- `ctp` - the ctp
- `layer` - the layer
- `paramName` - the param name
- `paramVal` - the param val

**Returns:**
- either null or an error message
setManagedElement

public java.lang.String setManagedElement(java.lang.String emsName,
java.lang.String meName,
java.lang.String subnetName,
java.lang.String userLabel,
java.lang.String location,
java.lang.String version,
java.lang.String productName,
java.lang.String communicationState,
java.lang.String nativeEMSName,
java.lang.String emsInSyncState,
java.lang.String owner,
java.lang.String[] supportedRates,
java.lang.String[] additionalInfo)

Creates a new ManagedElement.

Parameters:

emsName - the ems name
meName - the me name
subnetName - the subnet name
userLabel - the user label
location - the location
version - the version
productName - the product name
communicationState - the communication state
nativeEMSName - the native ems name
emsInSyncState - the ems in sync state
owner - the owner
supportedRates - the supported rates
additionalInfo - the additional info

Returns:

either null or an error message
setNT_Alarm

public java.lang.String setNT_Alarm(java.lang.String notificationId,
                                    java.lang.String ems,
                                    java.lang.String me,
                                    java.lang.String ptp,
                                    java.lang.String ctp,
                                    java.lang.String nativeEmsName,
                                    java.lang.String objectType,
                                    java.lang.String objectTypeQualifier,
                                    java.lang.String emsTime,
                                    java.lang.String neTime,
                                    java.lang.String isClearable,
                                    java.lang.String layerRate,
                                    java.lang.String perceivedSeverity,
                                    java.lang.String acknowledgeIndication,
                                    java.lang.String nativeProbableCause,
                                    java.lang.String probableCause,
                                    java.lang.String probableCauseQualifier,
                                    java.lang.String serviceAffecting,
                                    java.lang.String additionalText,
                                    java.lang.String rcaiIndicator,
                                    java.lang.String[] x733AdditionalInfo,
                                    java.lang.String x733BackedUpStatus,
                                    java.lang.String x733BackupObject,
                                    java.lang.String[] x733CorrelatedNotifications,
                                    java.lang.String x733EventType,
                                    java.lang.String[] x733MonitoredAttributes,
                                    java.lang.String[] x733ProposedRepairActions,
                                    java.lang.String[] x733SpecificProblems,
                                    java.lang.String[] x733TrendIndication,
                                    java.lang.String[] affectedTPList)

Creates a new NT_Alarm.

Parameters:

notificationId - the notification id
ems - the ems
me - the me
ptp - the ptp
ctp - the ctp
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
layerRate - the layer rate
perceivedSeverity - the perceived severity
acknowledgeIndication - the acknowledge indication
nativeProbableCause - the native probable cause
probableCause - the probable cause
probableCauseQualifier - the probable cause qualifier
serviceAffecting - the service affecting
additionalText - the additional text
rcaiIndicator - the rcai indicator
x733AdditionalInfo - the x733 additional info
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733CorrelatedNotifications - the x733 correlated notifications
x733EventType - the x733 event type
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733SpecificProblems - the x733 specific problems
x733TrendIndication - the x733 trend indication
affectedTPList - the affected tp list

**Returns:**

either null or an error message
setNT_TCA

public java.lang.String setNT_TCA(String notificationId,
                                     String ems,
                                     String level2Obj,
                                     String level3Obj,
                                     String level4Obj,
                                     String nativeEmsName,
                                     String objectType,
                                     String objectTypeQualifier,
                                     String emsTime,
                                     String neTime,
                                     String isClearable,
                                     String perceivedSeverity,
                                     String layerRate,
                                     String granularity,
                                     String pmParameterName,
                                     String pmLocation,
                                     String thresholdType,
                                     String floatValue,
                                     String unit,
                                     String acknowledgeIndication)

Creates a new NT_TCA.

Parameters:

- notificationId - the notification id
- ems - the ems
- level2Obj - the level2 obj
- level3Obj - the level3 obj
- level4Obj - the level4 obj
- nativeEmsName - the native ems name
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- perceivedSeverity - the perceived severity
- layerRate - the layer rate
- granularity - the granularity
- pmParameterName - the pm parameter name
- pmLocation - the pm location
- thresholdType - the threshold type
- floatValue - the float value
- unit - the unit
- acknowledgeIndication - the acknowledge indication

Returns:

either null or an error message
setPTP

public java.lang.String setPTP(java.lang.String ems,
        java.lang.String me,
        java.lang.String ptp,
        java.lang.String userLabel,
        java.lang.String nativeEMSName,
        java.lang.String owner,
        java.lang.String type,
        java.lang.String connectionState,
        java.lang.String tpMappingMode,
        java.lang.String direction,
        java.lang.String tpProtectionAssociation,
        java.lang.String edgePoint,
        java.lang.String ingressName,
        java.lang.String egressName,
        java.lang.String[] addInfo)

Creates a new PTP.

Parameters:
    ems - the ems
    me - the me
    ptp - the ptp
    userLabel - the user label
    nativeEMSName - the native ems name
    owner - the owner
    type - the type
    connectionState - the connection state
    tpMappingMode - the tp mapping mode
    direction - the direction
    tpProtectionAssociation - the tp protection association
    edgePoint - the edge point
    ingressName - the ingress name
    egressName - the egress name
    addInfo - the add info

Returns:
    either null or an error message
**setTopLevelSubnetwork**

```java
public java.lang.String setTopLevelSubnetwork(java.lang.String ems,
                                           java.lang.String multiLayerSN,
                                           java.lang.String owner,
                                           java.lang.String userLabel,
                                           java.lang.String nativeEMSName,
                                           java.lang.String subnetworkType,
                                           java.lang.String[] shorts,
                                           java.lang.String[] addInfo)
```

Creates a new TopLevelSubnetwork.

**Parameters:**
- `ems`: the ems
- `multiLayerSN`: the multi layer sn
- `owner`: the owner
- `userLabel`: the user label
- `nativeEMSName`: the native ems name
- `subnetworkType`: the subnetwork type
- `shorts`: the shorts
- `addInfo`: the add info

**Returns:**
- either null or an error message
public java.lang.String setTopologicalLink(java.lang.String ems,
java.lang.String tlName,
java.lang.String userLabel,
java.lang.String nativeEmsName,
java.lang.String owner,
java.lang.String direction,
java.lang.String rate,
java.lang.String node1Ems,
java.lang.String node1Me,
java.lang.String node1ptp,
java.lang.String node1ctp,
java.lang.String node2Ems,
java.lang.String node2Me,
java.lang.String node2ptp,
java.lang.String node2ctp,
java.lang.String[] addInfo)

Creates a new TopologicalLink.

Parameters:

ems - the ems
tlName - the tl name
userLabel - the user label
nativeEmsName - the native ems name
owner - the owner
direction - the direction
rate - the rate
node1Ems - the node1 ems
node1Me - the node1 me
node1ptp - the node1ptp
node1ctp - the node1ctp
node2Ems - the node2 ems
node2Me - the node2 me
node2ptp - the node2ptp
node2ctp - the node2ctp
addInfo - the add info

Returns:

either null or an error message
updateCTP

public void updateCTP(java.lang.String emsName,
                java.lang.String meName,
                java.lang.String ptpName,
                java.lang.String ctpName,
                java.lang.String newCTPName,
                java.lang.String userLabel,
                java.lang.String nativeEMSName,
                java.lang.String owner,
                java.lang.String type,
                java.lang.String connectionState,
                java.lang.String tpMappingMode,
                java.lang.String direction,
                java.lang.String tpProtectionAssociation,
                java.lang.String edgePoint,
                java.lang.String ingressName,
                java.lang.String egressName,
                java.lang.String[] transmissionParameters,
                java.lang.String[] additionalInfo)

Changes attributes for the given CTP. null values leave the attributes unchanged.

Parameters:

emsName - the ems name
meName - the me name
ptpName - the ptp name
ctpName - the ctp name
newCTPName - the new ctp name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
type - the type
connectionState - the connection state
tpMappingMode - the tp mapping mode
direction - the direction
tpProtectionAssociation - the tp protection association
edgePoint - the edge point
ingressName - the ingress name
egressName - the egress name
transmissionParameters - the transmission parameters
additionalInfo - the additional info
updateEMS

public void updateEMS(java.lang.String emsName,
                        java.lang.String newEmsName,
                        java.lang.String userLabel,
                        java.lang.String nativeEMSName,
                        java.lang.String owner,
                        java.lang.String emsVersion,
                        java.lang.String type,
                        java.lang.String[] additionalInfo)

Changes EMS attributes. null values leave the attributes unchanged.

Parameters:

emsName - The name of the EMS you are changing
newEmsName - the new ems name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
emsVersion - the ems version
type - the type
additionalInfo - the additional info

updateManagedElement

public void updateManagedElement(java.lang.String emsName,
                                 java.lang.String oldMeName,
                                 java.lang.String newMeName,
                                 java.lang.String subnetName,
                                 java.lang.String userLabel,
                                 java.lang.String location,
                                 java.lang.String version,
                                 java.lang.String productName,
                                 java.lang.String communicationState,
                                 java.lang.String nativeEMSName,
                                 java.lang.String emsInSyncState,
                                 java.lang.String owner,
                                 java.lang.String[] supportedRates,
                                 java.lang.String[] additionalInfo)

Changes attributes for the given Managed Element. null values leave the attributes unchanged.

Parameters:

emsName - the ems name
oldMeName - The name of the Managed Element you are changing.
newMeName - Can be used to give the Managed Element a new name.
subnetName - the subnet name
userLabel - the user label
location - the location
version - the version
productName - the product name
communicationState - the communication state
nativeEMSName - the native ems name
emsInSyncState - the ems in sync state
owner - the owner
supportedRates - the supported rates
additionalInfo - the additional info
public void updateMultiLayerSubnetwork(java.lang.String level1Object,
java.lang.String level2Object,
java.lang.String newLevel2Object,
java.lang.String userLabel,
java.lang.String owner,
java.lang.String nativeEmsName,
java.lang.String subnetworkType,
java.lang.String[] layerRate,
java.lang.String[] additionalInfo)

Changes attributes for the given MultiLayerSubnetwork. null values leave the attributes unchanged.

**Parameters:**

- level1Object - the level1 object
- level2Object - The name of the Subnetwork you are changing.
- newLevel2Object - Can be used to give the Subnetwork a new name.
- userLabel - the user label
- owner - the owner
- nativeEmsName - the native ems name
- subnetworkType - the subnetwork type
- layerRate - the layer rate
- additionalInfo - the additional info
public java.lang.String updateNTAlarm(java.lang.String notificationID,
java.lang.String level1Object,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String layerRate,
java.lang.String perceivedSeverity,
java.lang.String acknowledgeIndication,
java.lang.String nativeProbableCause,
java.lang.String probableCause,
java.lang.String probableCauseQualifier,
java.lang.String serviceAffecting,
java.lang.String additionalText,
java.lang.String rcaiIndicator,
java.lang.String[] x733AdditionalInfo,
java.lang.String x733BackedUpStatus,
java.lang.String x733BackupObject,
java.lang.String[] x733CorrelatedNotifications,
java.lang.String x733EventType,
java.lang.String[] x733MonitoredAttributes,
java.lang.String[] x733ProposedRepairActions,
java.lang.String[] x733SpecificProblems,
java.lang.String x733TrendIndication,
java.lang.String[] affectedTPList)

Update nt alarm.

Parameters:

notificationID - the notification id
level1Object - the level1 object
level2Object - the level2 object
level3Object - the level3 object
level4Object - the level4 object
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
layerRate - the layer rate
perceivedSeverity - the perceived severity
acknowledgeIndication - the acknowledge indication
nativeProbableCause - the native probable cause
probableCause - the probable cause
probableCauseQualifier - the probable cause qualifier
serviceAffecting - the service affecting
additionalText - the additional text
rcaiIndicator - the rcai indicator
x733AdditionalInfo - the x733 additional info
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733CorrelatedNotifications - the x733 correlated notifications
x733EventType - the x733 event type
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733SpecificProblems - the x733 specific problems
x733TrendIndication - the x733 trend indication
affectedTPList - the affected tp list

Returns:

the string
updateNTTCA

public java.lang.String updateNTTCA(java.lang.String notificationId,
java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String perceivedSeverity,
java.lang.String layerRate,
java.lang.String granularity,
java.lang.String pmParameterName,
java.lang.String pmLocation,
java.lang.String thresholdType,
java.lang.String value,
java.lang.String unit,
java.lang.String acknowledgeIndication)

Update nttca.

Parameters:

notificationId - the notification id
ems - the ems
level2Object - the level2 object
level3Object - the level3 object
level4Object - the level4 object
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
perceivedSeverity - the perceived severity
layerRate - the layer rate
granularity - the granularity
pmParameterName - the pm parameter name
pmLocation - the pm location
thresholdType - the threshold type
value - the value
unit - the unit
acknowledgeIndication - the acknowledge indication

Returns:

the string
public void updatePTP(java.lang.String emsName,
               java.lang.String meName,
               java.lang.String PTPName,
               java.lang.String newPTPName,
               java.lang.String userLabel,
               java.lang.String nativeEMSName,
               java.lang.String owner,
               java.lang.String type,
               java.lang.String connectionState,
               java.lang.String tpMappingMode,
               java.lang.String direction,
               java.lang.String tpProtectionAssociation,
               java.lang.String edgePoint,
               java.lang.String ingressName,
               java.lang.String egressName,
               java.lang.String[] transmissionParameters,
               java.lang.String[] additionalInfo)
Changes attributes for the given PTP/FTP. null values leave the attributes unchanged.

Parameters:

  emsName - the ems name
  meName - the me name
  PTPName - the pTP name
  newPTPName - Can be used to give the Managed Element a new name.
  userLabel - the user label
  nativeEMSName - the native ems name
  owner - the owner
  type - the type
  connectionState - the connection state
  tpMappingMode - the tp mapping mode
  direction - the direction
  tpProtectionAssociation - the tp protection association
  edgePoint - the edge point
  ingressName - the ingress name
  egressName - the egress name
  transmissionParameters - the transmission parameters
  additionalInfo - the additional info
updateTopologicalLink

public void updateTopologicalLink(java.lang.String emsName,
java.lang.String oldTlName,
java.lang.String newTlName,
java.lang.String userLabel,
java.lang.String nativeEmsName,
java.lang.String owner,
java.lang.String direction,
java.lang.String rate,
java.lang.String node1Ems,
java.lang.String node1Me,
java.lang.String node1ptp,
java.lang.String node1ctp,
java.lang.String node2Ems,
java.lang.String node2Me,
java.lang.String node2ptp,
java.lang.String node2ctp,
java.lang.String[] additionalInfo)

Update topological link.

Parameters:

emsName - the ems name
oldTlName - the old tl name
newTlName - the new tl name
userLabel - the user label
nativeEmsName - the native ems name
owner - the owner
direction - the direction
rate - the rate
node1Ems - the node1 ems
node1Me - the node1 me
node1ptp - the node1ptp
node1ctp - the node1ctp
node2Ems - the node2 ems
node2Me - the node2 me
node2ptp - the node2ptp
node2ctp - the node2ctp
additionalInfo - the additional info

com.ericsson.eos.interfaces

Interface DebugInterface

< Methods >

public interface DebugInterface

The Interface DebugInterface.

Methods
**setDebugToFile**

public void setDebugToFile(boolean bool)

Sets the debug to file.

**Parameters:**

bool - the new debug to file

---

**setDebugToScreen**

public void setDebugToScreen(boolean bool)

Sets the debug to screen.

**Parameters:**

bool - the new debug to screen

---

**write**

public void write(java.lang.String message, java.lang.Object from, java.util.logging.Level level)

Write.

**Parameters:**

message - the message
from - the from
level - the level

---

**com.ericsson.eos.interfaces**

**Interface NBI**

< Methods >

---

public interface NBI

This interface should be implemented by all classes that want to use the simulator for north-bound communication.

**Methods**
acknowledgeAlarms
public java.util.ArrayList acknowledgeAlarms(java.util.ArrayList alarms, java.util.ArrayList addInfo)

   Acknowledge alarms.
   Parameters:
      alarms - the alarms
      addInfo - the add info
   Returns:
      the array list

debug
public void debug(java.lang.String str, java.lang.Object o, java.util.logging.Level level)

   Debug.
   Parameters:
      str - the str
      o - the o
      level - the level

getActive
public boolean getActive()

   Gets the active.
   Returns:
      the active

getAdditionalInfo
public java.util.ArrayList getAdditionalInfo(int id)

   Gets the additional info.
   Parameters:
      id - the id
   Returns:
      the additional info
getAlarms

public java.util.ArrayList getAlarms(java.lang.String[] severity, java.lang.String[] probableCause)

Gets the alarms.

Parameters:
severity - the severity
probableCause - the probable cause

Returns:
the alarms

getAllAlarms

public java.lang.String[] getAllAlarms()

Gets the all alarms.

Returns:
the all alarms

getAllManagedElements

public java.util.ArrayList getAllManagedElements()

Gets the all managed elements.

Returns:
the all managed elements

getAllManagedElements

public java.util.ArrayList getAllManagedElements(java.lang.String ems, java.lang.String subnetwork)

Gets the all managed elements.

Parameters:
ems - the ems
subnetwork - the subnetwork

Returns:
the all managed elements
**getAllNodes**

**public java.lang.String[] getAllNodes()**

- Gets the all nodes.
- **Returns:**
  - the all nodes

---

**getAllTL**

**public java.lang.String[] getAllTL()**

- Gets the all tl.
- **Returns:**
  - the all tl

---

**getAllTopLevelSubnetworkLayerRates**

**public java.util.ArrayList getAllTopLevelSubnetworkLayerRates(int ems, java.lang.String subnet)**

- Gets the all top level subnetwork layer rates.
- **Parameters:**
  - ems - the ems
  - subnet - the subnet
- **Returns:**
  - the all top level subnetwork layer rates

---

**getAllTopLevelSubnetworks**

**public java.util.ArrayList getAllTopLevelSubnetworks()**

- Gets the all top level subnetworks.
- **Returns:**
  - the all top level subnetworks
**getCTP**

```java
public java.util.HashMap getCTP(java.lang.String ems,
            java.lang.String me,
            java.lang.String ptp,
            java.lang.String ctp)
```

Gets the cTP.

**Parameters:**
- ems - the ems
- me - the me
- ptp - the ptp
- ctp - the ctp

**Returns:**
- the cTP

---

**getContainedCurrentTPNames**

```java
public java.lang.String[] getContainedCurrentTPNames(java.lang.String ems,
            java.lang.String me,
            java.lang.String tp,
            short[] layerRate)
```

Gets the contained current tp names.

**Parameters:**
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**
- the contained current tp names

---

**getContainedCurrentTPs**

```java
public java.util.ArrayList getContainedCurrentTPs(java.lang.String ems,
            java.lang.String me,
            java.lang.String tp,
            short[] layerRate)
```

Gets the contained current tps.

**Parameters:**
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**
- the contained current tps
getContainedInUseTPNames
public java.lang.String[] getContainedInUseTPNames(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

Gets the contained in use tp names.

Parameters:
ems - the ems
me - the me
tp - the tp
layerRate - the layer rate

Returns:
the contained in use tp names

getContainedInUseTPs
public java.util.ArrayList getContainedInUseTPs(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

Gets the contained in use tps.

Parameters:
ems - the ems
me - the me
tp - the tp
layerRate - the layer rate

Returns:
the contained in use tps

getContainedPotentialTPNames
public java.lang.String[] getContainedPotentialTPNames(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

Gets the contained potential tp names.

Parameters:
ems - the ems
me - the me
tp - the tp
layerRate - the layer rate

Returns:
the contained potential tp names
getContainedPotentialTPs

public java.util.ArrayList getContainedPotentialTPs(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the contained potential t ps.

Parameters:
  ems - the ems
  me - the me
  tp - the tp
  layerRate - the layer rate

Returns:
  the contained potential t ps

getEms

public java.util.HashMap getEms()

Gets the ems.

Returns:
  the ems

getInterfaceName

public java.lang.String getInterfaceName()

This function should return the name that is associated with the NBI implementation, preferably the name of the protocol. It will be used when displaying the different NBI:s in the GUI, and when to choose which that should be active.

Returns:
  The name of the NBI implementation.

getLayerParameters

public java.util.ArrayList getLayerParameters(int objectID)

Gets the layer parameters.

Parameters:
  objectID - the object id

Returns:
  the layer parameters
**getManagedElement**

```java
public java.util.HashMap getManagedElement(java.lang.String ems, java.lang.String me)
```

Gets the managed element.

**Parameters:**
- ems - the ems
- me - the me

**Returns:**
- the managed element

---

**getNodes**

```java
public java.lang.String[] getNodes(java.lang.String node)
```

Gets the nodes.

**Parameters:**
- node - the node

**Returns:**
- the nodes

---

**getObjectName**

```java
public java.util.HashMap getObjectName(int objectId)
```

Gets the object name.

**Parameters:**
- objectId - the object id

**Returns:**
- the object name
getPTP

public java.util.HashMap getPTP(java.lang.String ems, java.lang.String me, java.lang.String ptp)

Gets the pTP.

Parameters:
  ems - the ems
  me - the me
  ptp - the ptp

Returns:
  the pTP

getPTPs

public java.util.ArrayList getPTPs(java.lang.String ems, java.lang.String me)

Gets the pTps.

Parameters:
  ems - the ems
  me - the me

Returns:
  the pTps

getSubnetwork

public java.util.HashMap getSubnetwork(java.lang.String ems, java.lang.String mlsn)

Gets the subnetwork.

Parameters:
  ems - the ems
  mlsn - the mlsn

Returns:
  the subnetwork
getSubnodes

public java.lang.String[] getSubnodes(java.lang.String node)

    Gets the subnodes.
    Parameters:
    node - the node
    Returns:
    the subnodes

getSupportedRates

public java.util.HashMap getSupportedRates(int id)

    Gets the supported rates.
    Parameters:
    id - the id
    Returns:
    the supported rates

getTL

public java.lang.String[] getTL(java.lang.String node)

    Gets the tL.
    Parameters:
    node - the node
    Returns:
    the tL

getTopNode

public java.lang.String getTopNode()

    Gets the top node.
    Returns:
    the top node
getX733AdditionalInfo

public java.util.ArrayList getX733AdditionalInfo(java.lang.String id)

Gets the x733 additional info.

Parameters:

id - the id

Returns:

the x733 additional info

getX733MonitoredAttribute

public java.util.ArrayList getX733MonitoredAttribute(java.lang.String id)

Gets the x733 monitored attribute.

Parameters:

id - the id

Returns:

the x733 monitored attribute

notificate

public void notificate(org.omg.CosNotification.StructuredEvent se)

This function sends the alarm that is given through the channel that setupNotificationRoute() setups. It is up to the receiving end to handle the alarm after it has passed through the NBI.

Parameters:

se - the se

sendAlarm

public void sendAlarm(java.util.HashMap alarm)

Send alarm.

Parameters:

alarm - the alarm
**sendAttributeChanged**

```java
public void sendAttributeChanged(java.lang.String emsName,
                                java.lang.String level2Object,
                                java.lang.String level3Object,
                                java.lang.String level4Object,
                                java.lang.String notificationId,
                                java.lang.String objectType,
                                java.lang.String objectTypeQualifier,
                                java.lang.String emsTime,
                                java.lang.String neTime,
                                java.lang.String edgePointRelated,
                                java.lang.String[] attributList)
```

Send attribute changed.

**Parameters:**
- `emsName` - the ems name
- `level2Object` - the level2 object
- `level3Object` - the level3 object
- `level4Object` - the level4 object
- `notificationId` - the notification id
- `objectType` - the object type
- `objectTypeQualifier` - the object type qualifier
- `emsTime` - the ems time
- `neTime` - the ne time
- `edgePointRelated` - the edge point related
- `attributList` - the attribut list

---

**sendHeartBeat**

```java
public void sendHeartBeat(java.lang.String ems,
                         java.lang.String me,
                         java.lang.String notificationID,
                         java.lang.String emsTime)
```

Send heart beat.

**Parameters:**
- `ems` - the ems
- `me` - the me
- `notificationID` - the notification id
- `emsTime` - the ems time
sendStateChanged

public void sendStateChanged(java.lang.String emsName,
                             java.lang.String level2Object,
                             java.lang.String level3Object,
                             java.lang.String level4Object,
                             java.lang.String notificationId,
                             java.lang.String objectType,
                             java.lang.String objectTypeQualifier,
                             java.lang.String emsTime,
                             java.lang.String neTime,
                             java.lang.String edgePointRelated,
                             java.lang.String[] attributList)

Send state changed.

Parameters:
- emsName - the ems name
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object
- notificationId - the notification id
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- edgePointRelated - the edge point related
- attributList - the attribut list

setActive

public void setActive(boolean bool)

Sets the active.

Parameters:
- bool - the new active

setModel

public void setModel(Model model)

Sets the model.

Parameters:
- model - the new model
setupNotificationRoute

public void setupNotificationRoute()

To be able to use notification, the implementation of NBI have to setup the communication to the receiving end on it's own. This function will be run at startup for each found implementation of NBI and will thereby secure a channel for forwarding alarms over the north bound interface.

unacknowledgeAlarms

public java.util.ArrayList unacknowledgeAlarms(java.util.ArrayList alarms, java.util.ArrayList addInfo)

Unacknowledge alarms.

Parameters:
alarms - the alarms
addInfo - the add info

Returns:
the array list

updateEMS

public void updateEMS(java.lang.String emsName,
java.lang.String newEmsName,
java.lang.String userLabel,
java.lang.String nativeEMSName,
java.lang.String owner,
java.lang.String emsVersion,
java.lang.String type,
java.lang.String[] additionalInfo)

Update ems.

Parameters:
emsName - the ems name
newEmsName - the new ems name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
emsVersion - the ems version
type - the type
additionalInfo - the additional info
Package com.ericsson.eos.junitTests

Class Summary

AllTests

GetterTester

MySQLTest

com.ericsson.eos.junitTests

Class AllTests

java.lang.Object

|-- com.ericsson.eos.junitTests.AllTests

< Constructors > < Methods >

public class AllTests
extends java.lang.Object

Constructors

AllTests

public AllTests()

Methods

main

public static void main(java.lang.String[] args)

suite

public static junit.framework.Test suite()
com.ericsson.eos.junitTests

Class GetterTester

java.lang.Object
   +--com.ericsson.eos.junitTests.GetterTester

< Constructors >

public class GetterTester
extends java.lang.Object

Constructors

GetterTester

public GetterTester()
Methods

setUp

public void setUp()

Overrides:
setUp in class junit.framework.TestCase

tearDown

public void tearDown()

Overrides:
tearDown in class junit.framework.TestCase

testFilterCritical

public void testFilterCritical()

testFilterLOS

public void testFilterLOS()

testFilterWarning

public void testFilterWarning()

testNoCorrectFilter

public void testNoCorrectFilter()
Package com.ericsson.eos.model

Class Summary

**Model**
Model.java - Create a model object to get access to setters, getters, update and delete functions for TMF814 object types.

**ModelAlarm**
The Class ModelAlarm.

**ModelCtp**
The Class ModelCtp.

**ModelEms**
The Class ModelEms.

**ModelHelper**
The Class ModelHelper.

**ModelInit**
The Class ModelInit.

**ModelMe**
The Class ModelMe.

**ModelMlsn**
The Class ModelMlsn.

**ModelPtp**
The Class ModelPtp.

**ModelSelection**
The Class ModelSelection.

**ModelTca**
The Class ModelTca.

**ModelTI**
The Class ModelTI.

**MyTableModel**
The Class MyTableModel.
public class Model
extends ModelAlarm

Model.java - Create a model object to get access to setters, getters, update and delete functions for TMF814 object types. Part of the Model-View-Controller architecture for the GUI, and an access point to TMF814 objects for the Northbound Interface NBI. Model.java communicates with the database through the Database Interface DBI. Debug messages are created for all functions except for get functions.

Constructors

Model

public Model()

Instantiates a new model.
com.ericsson.eos.model

Class ModelAlarm

java.lang.Object
   +--java.util.Observable
      +--ModelInit
      +--ModelHelper
         +--ModelEms
            +--ModelMe
               +--ModelPtp
               +--ModelCtp
               +--ModelMlsn
               +--ModelTl
               +--ModelTca
      +--com.ericsson.eos.model.ModelAlarm

Direct Known Subclasses:
   Model

< Constructors > < Methods >

public class ModelAlarm
extends ModelTca

The Class ModelAlarm.

Constructors

ModelAlarm

public ModelAlarm()
acknowledgeAlarm

public java.lang.String acknowledgeAlarm(java.lang.String notificationId, java.lang.String acknowledgeIndication, java.lang.String[] x733AdditionalInfo)

Acknowledge or unacknowledge alarms.

Parameters:
- notificationId - the notification id
- acknowledgeIndication - the acknowledge indication
- x733AdditionalInfo - the x733 additional info

Returns:
true if acknowledge/unacknowledge succeeded.

deleteAlarms

public void deleteAlarms(java.util.ArrayList list)

Delete alarms.

Parameters:
- list - the list

getAlarm

public java.util.HashMap getAlarm(java.lang.String notificationID)

Gets the alarm.

Parameters:
- notificationID - the notification id

Returns:
the alarm

getAlarms

public java.util.ArrayList getAlarms(java.lang.String[] serverityFilter, java.lang.String[] probCauseFilter)

Retrieves all alarms filtered on severity and probableCause.

Parameters:
- serverityFilter - the serverity filter
- probCauseFilter - the prob cause filter

Returns:
an ArrayList of HashMaps containing the attributes and values of the filtered alarms.
getAllActiveAlarms

public java.util.ArrayList getAllActiveAlarms()

Retrieves all EMS and ME active alarms.

Returns:
    an ArrayList of HashMaps containing attributes and values.

getAllAlarmParameters

public java.util.ArrayList getAllAlarmParameters()

Retrieves all the attribute names of an alarm. used from the GUI.

Returns:
    an ArrayList of attribute names.

g getX733AdditionalInfo

public java.util.ArrayList getX733AdditionalInfo(java.lang.String id)

Retrieves X733 AdditionalInfo for the specified alarm.

Parameters:
    id - the id

Returns:
    the x733 additional info

g getX733MonitoredAttribute

public java.util.ArrayList getX733MonitoredAttribute(java.lang.String id)

Retrieves.

Parameters:
    id - the id

Returns:
    the x733 monitored attribute
setNT_ALARM

public java.lang.String setNT_ALARM(java.lang.String notificationId,
java.lang.String ems,
java.lang.String level2Obj,
java.lang.String level3Obj,
java.lang.String level4Obj,
java.lang.String nativeEMSName,
java.lang.String nativeProbableCause,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String clearable,
java.lang.String layerRate,
java.lang.String probableCause,
java.lang.String probableCauseQualifier,
java.lang.String perceivedSeverity,
java.lang.String serviceAffecting,
java.lang.String[] affectedTPList,
java.lang.String additionalText,
java.lang.String x733EventType,
java.lang.String[] x733SpecificProblems,
java.lang.String x733BackedUpStatus,
java.lang.String x733BackupObject,
java.lang.String x733TrendIndication,
java.lang.String[] x733CorrelatedNotifications,
java.lang.String[] x733MonitoredAttributes,
java.lang.String[] x733ProposedRepairActions,
java.lang.String[] x733AdditionalInfo,
java.lang.String rcaiIndicator,
java.lang.String acknowledgeIndication)

Creates a new NT_Alarm.

Parameters:

- notificationId - the notification id
- ems - the ems
- level2Obj - the level2 obj
- level3Obj - the level3 obj
- level4Obj - the level4 obj
- nativeEMSName - the native ems name
- nativeProbableCause - the native probable cause
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- clearable - the clearable
- layerRate - the layer rate
- probableCause - the probable cause
- probableCauseQualifier - the probable cause qualifier
- perceivedSeverity - the perceived severity
- serviceAffecting - the service affecting
- affectedTPList - the affected tp list
- additionalText - the additional text
- x733EventType - the x733 event type
- x733SpecificProblems - the x733 specific problems
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733TrendIndication - the x733 trend indication
x733CorrelatedNotifications - the x733 correlated notifications
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733AdditionalInfo - the x733 additional info
rcaiIndicator - the rcai indicator
acknowledgeIndication - the acknowledge indication

Returns:

the string
updateAlarm

public java.lang.String updateAlarm(java.lang.String notificationID,
                java.lang.String level1Obj,
                java.lang.String level2Obj,
                java.lang.String level3Obj,
                java.lang.String level4Obj,
                java.lang.String nativeEMSName,
                java.lang.String nativeProbableCause,
                java.lang.String objectType,
                java.lang.String objectTypeQualifier,
                java.lang.String emsTime,
                java.lang.String neTime,
                java.lang.String clearable,
                java.lang.String layerRate,
                java.lang.String probableCause,
                java.lang.String probableCauseQualifier,
                java.lang.String perceivedSeverity,
                java.lang.String serviceAffecting,
                java.lang.String[] affectedTPLList,
                java.lang.String additionalText,
                java.lang.String x733EventType,
                java.lang.String[] x733SpecificProblems,
                java.lang.String x733BackedUpStatus,
                java.lang.String x733BackupObject,
                java.lang.String x733TrendIndication,
                java.lang.String[] x733CorrelatedNotifications,
                java.lang.String[] x733MonitoredAttributes,
                java.lang.String[] x733ProposedRepairActions,
                java.lang.String[] x733AdditionalInfo,
                java.lang.String rcaiIndicator,
                java.lang.String acknowledgeIndication)

x733CorrelatedNotifications,

x733MonitoredAttributes,

x733ProposedRepairActions,

Change alarm attribute values.

Parameters:

notificationID - the notification id
level1Obj - the level1 obj
level2Obj - the level2 obj
level3Obj - the level3 obj
level4Obj - the level4 obj
nativeEMSName - the native ems name
nativeProbableCause - the native probable cause
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
clearable - the clearable
layerRate - the layer rate
probableCause - the probable cause
probableCauseQualifier - the probable cause qualifier
perceivedSeverity - the perceived severity
serviceAffecting - the service affecting
affectedTPLList - the affected tp list
additionalText - the additional text
x733EventType - the x733 event type
x733SpecificProblems - the x733 specific problems
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733TrendIndication - the x733 trend indication
x733CorrelatedNotifications - the x733 correlated notifications
x733MonitoredAttributes - the x733 monitored attributes
x733ProposedRepairActions - the x733 proposed repair actions
x733AdditionalInfo - the x733 additional info
rcaiIndicator - the rcai indicator
acknowledgeIndication - the acknowledge indication

Returns:
the string

com.ericsson.eos.model

Class ModelCtp

java.lang.Object
  ^---java.util.Observable
    ^--ModelInit
       ^---ModelHelper
          ^---ModelEms
             ^---ModelMe
                ^---ModelPtp
                   ^---com.ericsson.eos.model.ModelCtp

Direct Known Subclasses:
  ModelMlsn

public class ModelCtp
extends ModelPtp

The Class ModelCtp.

Constructors

ModelCtp

public ModelCtp()
createCTP

public void createCTP(java.lang.String emsName,
             java.lang.String meName,
             java.lang.String ptpName,
             java.lang.String ctpName,
             java.lang.String userLabel,
             java.lang.String nativeEMSName,
             java.lang.String owner,
             java.lang.String type,
             java.lang.String connectionState,
             java.lang.String tpMappingMode,
             java.lang.String direction,
             java.lang.String tpProtectionAssociation,
             java.lang.String edgePoint,
             java.lang.String ingressName,
             java.lang.String egressName,
             java.lang.String[] additionalInfo)

Create a new CTP.

Parameters:
  emsName - the ems name
  meName - the me name
  ptpName - the ptp name
  ctpName - the ctp name
  userLabel - the user label
  nativeEMSName - the native ems name
  owner - the owner
  type - the type
  connectionState - the connection state
  tpMappingMode - the tp mapping mode
  direction - the direction
  tpProtectionAssociation - the tp protection association
  edgePoint - the edge point
  ingressName - the ingress name
  egressName - the egress name
  additionalInfo - the additional info

deleteCTP

public void deleteCTP(java.lang.String ems,
              java.lang.String me,
              java.lang.String ptp,
              java.lang.String ctp)

Delete ctp.

Parameters:
  ems - the ems
  me - the me
  ptp - the ptp
  ctp - the ctp
getCTP

public java.util.HashMap getCTP(java.lang.String ems,
        java.lang.String me,
        java.lang.String ptp,
        java.lang.String ctp)

Gets the cTP.

Parameters:
    ems - the ems
    me - the me
    ptp - the ptp
    ctp - the ctp

Returns:
    the cTP

getContainedCurrentTPNames

public java.lang.String[] getContainedCurrentTPNames(java.lang.String ems,
        java.lang.String me,
        java.lang.String tp,
        short[] layerRate)

Get all Current TP names.

Parameters:
    ems - the ems
    me - the me
    tp - the tp
    layerRate - the layer rate

Returns:
    a list of names.

getContainedCurrentTPs

public java.util.ArrayList getContainedCurrentTPs(java.lang.String ems,
        java.lang.String me,
        java.lang.String tp,
        short[] layerRate)

Get all Current TPs.

Parameters:
    ems - the ems
    me - the me
    tp - the tp
    layerRate - the layer rate

Returns:
    an ArrayList of HashMaps containing the attributes and values.
getContainedInUseTPNames

public java.lang.String[] getContainedInUseTPNames(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Get all InUseTPNames.

Parameters:
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

Returns:
a list of names.

getContainedInUseTPs

public java.util.ArrayList getContainedInUseTPs(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Get all Contained InUse TPs.

Parameters:
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

Returns:
an ArrayList of HashMaps containing the attributes and values.

getContainedPotentialTPNames

public java.lang.String[] getContainedPotentialTPNames(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Get all Contained Potential TP Names.

Parameters:
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

Returns:
a list of names.
public java.util.ArrayList getContainedPotentialTPs(java.lang.String ems, java.lang.String me, java.lang.String tp, short[] layerRate)

Gets the contained potential TPs.

**Parameters:**
- ems - the ems
- me - the me
- tp - the tp
- layerRate - the layer rate

**Returns:**
- an ArrayList of HashMaps containing the attributes and values.
public void updateCTP (java.lang.String emsName, 
        java.lang.String meName, 
        java.lang.String ptpName, 
        java.lang.String ctpName, 
        java.lang.String newCTPName, 
        java.lang.String userLabel, 
        java.lang.String nativeEMSName, 
        java.lang.String owner, 
        java.lang.String type, 
        java.lang.String connectionState, 
        java.lang.String tpMappingMode, 
        java.lang.String direction, 
        java.lang.String tpProtectionAssociation, 
        java.lang.String edgePoint, 
        java.lang.String ingressName, 
        java.lang.String egressName, 
        java.lang.String[] layerParam, 
        java.lang.String[] additionalInfo)

Update ctp.

Parameters:

  emsName - the ems name
  meName - the me name
  ptpName - the ptp name
  ctpName - the ctp name
  newCTPName - the new ctp name
  userLabel - the user label
  nativeEMSName - the native ems name
  owner - the owner
  type - the type
  connectionState - the connection state
  tpMappingMode - the tp mapping mode
  direction - the direction
  tpProtectionAssociation - the tp protection association
  edgePoint - the edge point
  ingressName - the ingress name
  egressName - the egress name
  layerParam - the layer param
  additionalInfo - the additional info
Class ModelEms

java.lang.Object
    +--java.util.Observable
        +--ModelInit
            +--ModelHelper
                +--com.ericsson.eos.model.ModelEms

Direct Known Subclasses:
    ModelMe

< Constructors > < Methods >

public class ModelEms
extends ModelHelper

The Class ModelEms.

Constructors

ModelEms

public ModelEms ()

Methods

deleteEms

public void deleteEms (java.lang.String ems)

    Remove the EMS and all underlying objects.

    Parameters:
        ems - the ems

getEms

public java.util.HashMap getEms ()

    Retrieve the EMS information.

    Returns:
        a HashMap with attributes and values respectively.
getEmsName

public java.lang.String getEmsName()

Retrieve the EMS Name.

Returns:
A String representation of the EMS Name.

setEMS

public void setEMS(java.lang.String ems,
                   java.lang.String userLabel,
                   java.lang.String nativeEmsName,
                   java.lang.String owner,
                   java.lang.String emsVersion,
                   java.lang.String type,
                   java.lang.String[] additionalInfo)

Set the EMS information. Used by the parser during system initialization.

Parameters:
ems - the ems
userLabel - the user label
nativeEmsName - the native ems name
owner - the owner
emsVersion - the ems version
type - the type
additionalInfo - the additional info

updateEMS

public void updateEMS(java.lang.String emsName,
                       java.lang.String newEMSName,
                       java.lang.String userLabel,
                       java.lang.String nativeEMSName,
                       java.lang.String owner,
                       java.lang.String emsVersion,
                       java.lang.String type,
                       java.lang.String[] additionalInfo)

Change EMS attribute values. Notifies NBI about changes.

Parameters:
emsName - the ems name
newEMSName - the new ems name
userLabel - the user label
nativeEMSName - the native ems name
owner - the owner
emsVersion - the ems version
type - the type
additionalInfo - the additional info
**Class ModelHelper**

java.lang.Object  
   \|--java.util.Observable  
      \|--ModelInit  
         \|--com.ericsson.eos.model.ModelHelper

Direct Known Subclasses:
  ModelEms

< Constructors > < Methods >

public class ModelHelper
extends ModelInit

The Class ModelHelper.

**Constructors**

ModelHelper

public ModelHelper()

**Methods**

changing

public void changing(java.lang.String type)

   Changing.

Parameters:

   type - the type
createTransmissionParameters

public void createTransmissionParameters(java.lang.String emsName,
java.lang.String meName,
java.lang.String ptpName,
java.lang.String ctpName,
java.lang.String layer,
java.lang.String[] paramName,
java.lang.String[] paramValue)

Creates the transmission parameters.

Parameters:

emsName - the ems name
meName - the me name
ptpName - the ptp name
ctpName - the ctp name
layer - the layer
paramName - the param name
paramValue - the param value

getAdditionalInfo

public java.util.ArrayList getAdditionalInfo(int id)

Gets the additional info.

Parameters:

id - the id

Returns:

the additional info

getConf

public Config getConf()

Gets the conf.

Returns:

the conf

getDatabaseStatus

public boolean getDatabaseStatus()

Gets the database status.

Returns:

the database status
**getDebug**

```java
public Debugger getDebug()
```

Gets the debug.

**Returns:**
the debug

---

**getLayerParameters**

```java
public java.util.ArrayList getLayerParameters(int id)
```

Gets the layer parameters.

**Parameters:**
- id - the id

**Returns:**
the layer parameters

---

**getLayerRate**

```java
public java.lang.String getLayerRate(short s)
```

Gets the layer rate.

**Parameters:**
- s - the s

**Returns:**
the layer rate

---

**getNBI**

```java
public NBI getNBI(java.lang.String name)
```

Gets the nBI.

**Parameters:**
- name - the name

**Returns:**
the nBI
getNBIs

public java.util.ArrayList getNBIs()

Gets the nB is.

Returns:
the nB is

getObjectName

public java.util.HashMap getObjectName(int id)

Gets the object name.

Parameters:
  id - the id

Returns:
the object name

getObjectNameId

public int getObjectNameId(java.lang.String level1Object,
                          java.lang.String level2Object,
                          java.lang.String level3Object)

Gets the object name id.

Parameters:
  level1Object - the level1 object
  level2Object - the level2 object
  level3Object - the level3 object

Returns:
the object name id
**getObjectNamId**

```java
public int getObjectNameId(java.lang.String level1Object,
                          java.lang.String level2Object,
                          java.lang.String level3Object,
                          java.lang.String level4Object)
```

Gets the object name id.

**Parameters:**
- level1Object - the level1 object
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object

**Returns:**
- the object name id

---

**getServiceStatus**

```java
public boolean getServiceStatus()
```

Gets the service status.

**Returns:**
- the service status

---

**getSupportedRates**

```java
public java.util.HashMap getSupportedRates(int id)
```

Gets the supported rates.

**Parameters:**
- id - the id

**Returns:**
- the supported rates

---

**printHashMap**

```java
public void printHashMap(java.util.HashMap temp)
```

Prints the hash map.

**Parameters:**
- temp - the temp
resetDB

public void resetDB()
    Reset db.

setDebugger

public void setDebugger(Debugger debug)
    Sets the debugger.
    Parameters:
    debug - the new debugger

setServiceStatus

public void setServiceStatus(boolean bool)
    Sets the service status.
    Parameters:
    bool - the new service status

com.ericsson.eos.model

Class ModelInit

java.lang.Object
    +--java.util.Observable
        +--com.ericsson.eos.model.ModelInit

Direct Known Subclasses:
    ModelHelper

< Constructors >

public class ModelInit
extends java.util.Observable

The Class ModelInit.

Constructors
`ModelInit`

```java
public ModelInit()

    Instantiates a new model init.
```

`com.ericsson.eos.model`

**Class ModelMe**

```java
java.lang.Object
    +--java.util.Observable
        +--ModelInit
            +--ModelHelper
                +--ModelEms
                    +--com.ericsson.eos.model.ModelMe
```

Direct Known Subclasses:
- `ModelPtp`

**Constructors**

**ModelMe**

```java
public class ModelMe extends ModelEms

    The Class ModelMe.
```

**Methods**

**deleteME**

```java
public void deleteME(java.lang.String ems,
                      java.lang.String me)

    Delete the managed element with the give ems- and me-name.

    Parameters:
    ems - the ems
    me - the me
```
**getAllManagedElements**

public java.util.ArrayList getAllManagedElements()

Retrieve all Managed Elements.

**Returns:**

an ArrayList of HashMaps containing attributes and values.

---

**getAllManagedElements**

public java.util.ArrayList getAllManagedElements(java.lang.String ems,
java.lang.String subnetwork)

Retrieve all Managed elements under the given subnetwork.

**Parameters:**

ems - the ems
subnetwork - the subnetwork

**Returns:**

an ArrayList of HashMaps containing attributes and values.

---

**getManagedElement**

public java.util.HashMap getManagedElement(java.lang.String ems,
java.lang.String name)

Retrieve the managed Element with the given ems- and me-name.

**Parameters:**

ems - the ems
name - the name

**Returns:**

a Hashmap containing attributes and values.

---

**getMeStatus**

public int getMeStatus(java.lang.String node)

Retrieve the given managed element’s communication state status.

**Parameters:**

node - the node

**Returns:**

1 if the communication state of the managed element is CS_AVAILABLE, 0 if it is CS_UNAVAILABLE.
public void setManagedElement(java.lang.String emsName,
                        java.lang.String meName,
                        java.lang.String subnetName,
                        java.lang.String userLabel,
                        java.lang.String location,
                        java.lang.String version,
                        java.lang.String productName,
                        java.lang.String communicationState,
                        java.lang.String nativeEmsName,
                        java.lang.String emsInSyncState,
                        java.lang.String owner,
                        java.lang.String[] supportedRates,
                        java.lang.String[] additionalInfo)

Create a new ManagedElement.

Parameters:

- emsName - the ems name
- meName - the me name
- subnetName - the subnet name
- userLabel - the user label
- location - the location
- version - the version
- productName - the product name
- communicationState - the communication state
- nativeEmsName - the native ems name
- emsInSyncState - the ems in sync state
- owner - the owner
- supportedRates - the supported rates
- additionalInfo - the additional info
updateManagedElement

public void updateManagedElement(java.lang.String emsName,
                               java.lang.String oldMeName,
                               java.lang.String newMeName,
                               java.lang.String subnetName,
                               java.lang.String userLabel,
                               java.lang.String location,
                               java.lang.String version,
                               java.lang.String productName,
                               java.lang.String communicationState,
                               java.lang.String nativeEMSName,
                               java.lang.String emsInSyncState,
                               java.lang.String owner,
                               java.lang.String[] supportedRates,
                               java.lang.String[] additionalInfo)

Change ManagedElement attribute values. Notifies NBI about changes.

Parameters:

- emsName - the ems name
- oldMeName - the old me name
- newMeName - the new me name
- subnetName - the subnet name
- userLabel - the user label
- location - the location
- version - the version
- productName - the product name
- communicationState - the communication state
- nativeEMSName - the native ems name
- emsInSyncState - the ems in sync state
- owner - the owner
- supportedRates - the supported rates
- additionalInfo - the additional info
public class ModelMlsn
extends ModelCtp

The Class ModelMlsn.

Constructors

ModelMlsn

public ModelMlsn()

Methods

deleteMLSN

public void deleteMLSN(java.lang.String ems, java.lang.String mlsn)

Removes the multiLayerSubnetwork.

Parameters:
  ems - the ems
  mlsn - the mlsn
getAllTopLevelSubnetworks

public java.util.ArrayList getAllTopLevelSubnetworks()

Retrieve all MultiLayerSubnetworks.

Returns:
an ArrayList of HashMaps containing attributes and values.

getSubNodes

public java.util.ArrayList getSubNodes(java.lang.String ems)

Retrieves all the MultiLayerSubnetworkNames under the given ems.

Parameters:
ems - the ems

Returns:
an ArrayList of names.

getSubnetwork

public java.util.HashMap getSubnetwork(java.lang.String ems, java.lang.String mlsn)

Retrieves the MultiLayerSubnetwork with the given ems- and subnetwork-name.

Parameters:
ems - the ems
mlsn - the mlsn

Returns:
a HashMap containing attributes and values.
setMultiLayerSubnetwork

public void setMultiLayerSubnetwork(java.lang.String ems,
                                     java.lang.String multiLayerSN,
                                     java.lang.String owner,
                                     java.lang.String userLabel,
                                     java.lang.String nativeEmsName,
                                     java.lang.String subnetworkType,
                                     java.lang.String[] layerRates,
                                     java.lang.String[] additionalInfo)

Create a new MultiLayerSubnetwork.

Parameters:

ems - the ems
multiLayerSN - the multi layer sn
owner - the owner
userLabel - the user label
nativeEmsName - the native ems name
subnetworkType - the subnetwork type
layerRates - the layer rates
additionalInfo - the additional info

updateMultiLayerSubnetwork

public void updateMultiLayerSubnetwork(java.lang.String ems,
                                       java.lang.String subnet,
                                       java.lang.String newSubnet,
                                       java.lang.String userLabel,
                                       java.lang.String owner,
                                       java.lang.String nativeEmsName,
                                       java.lang.String subnetworkType,
                                       java.lang.String[] layerRate,
                                       java.lang.String[] additionalInfo)

Change MultiLayerSubnetwork attribute values. Notifies NBI about changes.

Parameters:

ems - the ems
subnet - the subnet
newSubnet - the new subnet
userLabel - the user label
owner - the owner
nativeEmsName - the native ems name
subnetworkType - the subnetwork type
layerRate - the layer rate
additionalInfo - the additional info
public class ModelPtp extends ModelMe

The Class ModelPtp.

### Constructors

**ModelPtp**

class ModelPtp()

### Methods

**deletePTP**

public void deletePTP(java.lang.String ems,
                        java.lang.String me,
                        java.lang.String ptp)

Delete the PTP/FTP.

**Parameters:**

- ems - the ems
- me - the me
- ptp - the ptp
getPTP

public java.util.HashMap getPTP(java.lang.String ems, java.lang.String me, java.lang.String ptp)

Get PTP/FTP information.

Parameters:
ems - the ems
me - the me
ptp - the ptp

Returns:
a HashMap containing attributes and values.

getPTPNames

public java.lang.String[] getPTPNames(java.lang.String ems, java.lang.String me)

Retrieve all PTP/FTP Names.

Parameters:
ems - the ems
me - the me

Returns:
a list of Strings

getPTPs

public java.util.ArrayList getPTPs(java.lang.String ems, java.lang.String me)

Retrieve all PTP's and FTP's under the given ems- and an me- name.

Parameters:
ems - the ems
me - the me

Returns:
an ArrayList of HashMaps containing attributes and values.
public void setPTP(java.lang.String emsName,
                 java.lang.String meName,
                 java.lang.String ptpName,
                 java.lang.String userLabel,
                 java.lang.String nativeEMSName,
                 java.lang.String owner,
                 java.lang.String type,
                 java.lang.String connectionState,
                 java.lang.String tpMappingMode,
                 java.lang.String direction,
                 java.lang.String tpProtectionAssociation,
                 java.lang.String edgePoint,
                 java.lang.String ingressName,
                 java.lang.String egressName,
                 java.lang.String[] additionalInfo)

Creates a new PTP/FTP.

**Parameters:**

- emsName - the ems name
- meName - the me name
- ptpName - the ptp name
- userLabel - the user label
- nativeEMSName - the native ems name
- owner - the owner
- type - the type
- connectionState - the connection state
- tpMappingMode - the tp mapping mode
- direction - the direction
- tpProtectionAssociation - the tp protection association
- edgePoint - the edge point
- ingressName - the ingress name
- egressName - the egress name
- additionalInfo - the additional info
public void updatePTP(java.lang.String emsName,
    java.lang.String meName,
    java.lang.String ptpName,
    java.lang.String newPTPName,
    java.lang.String userLabel,
    java.lang.String nativeEMSName,
    java.lang.String owner,
    java.lang.String type,
    java.lang.String connectionState,
    java.lang.String tpMappingMode,
    java.lang.String direction,
    java.lang.String tpProtectionAssociation,
    java.lang.String edgePoint,
    java.lang.String ingressName,
    java.lang.String egressName,
    java.lang.String[] layerParam,
    java.lang.String[] additionalInfo)

Change PTP/FTP attribute values. Notifies NBI about changes.

Parameters:

  emsName - the ems name
  meName - the me name
  ptpName - the ptp name
  newPTPName - the new ptp name
  userLabel - the user label
  nativeEMSName - the native ems name
  owner - the owner
  type - the type
  connectionState - the connection state
  tpMappingMode - the tp mapping mode
  direction - the direction
  tpProtectionAssociation - the tp protection association
  edgePoint - the edge point
  ingressName - the ingress name
  egressName - the egress name
  layerParam - the layer param
  additionalInfo - the additional info

com.ericsson.eos.model

Class ModelSelection

java.lang.Object
    +--java.util.Observable
        +--com.ericsson.eos.model.ModelSelection

< Constructors > < Methods >
The Class ModelSelection.

**Constructors**

**ModelSelection**

```java
public ModelSelection(java.lang.String type)

    Instantiates a new model selection.

    Parameters:

    type - the type
```

**Methods**

**getLastSelected**

```java
public java.lang.String getLastSelected()

    Gets the last selected.

    Returns:

    the last selected
```

**getLastSelectedLevel**

```java
public int getLastSelectedLevel()

    Gets the last selected level.

    Returns:

    the last selected level
```

**getSelectedPaths**

```java
public java.lang.String[] getSelectedPaths()

    Gets the selected paths.

    Returns:

    the selected paths
```
**setLastSelected**

```java
public void setLastSelected(java.lang.String lastSelected, int level)
```

Sets the last selected.

**Parameters:**
- `lastSelected` - the last selected
- `level` - the level

---

**setSelectedPaths**

```java
public void setSelectedPaths(java.lang.String[] selectedPaths)
```

Sets the selected paths.

**Parameters:**
- `selectedPaths` - the new selected paths

---

**com.ericsson.eos.model**

**Class ModelTca**

```java
java.lang.Object
  +--java.util.Observable
    +--ModelInit
      +--ModelHelper
        +--ModelEms
          +--ModelMe
            +--ModelPtp
              +--ModelCtp
                +--ModelMlsn
                  +--ModelTl
                    +--com.ericsson.eos.model.ModelTca
```

**Direct Known Subclasses:**
- `ModelAlarm`

---

< [Constructors] > < [Methods] >

**public class ModelTca**

extends `ModelTl`
The Class ModelTca.

Constructors

**ModelTca**

```java
public ModelTca()
```

Methods
createNT_TCA
public void createNT_TCA(java.lang.String notificationId,
    java.lang.String ems,
    java.lang.String level2Obj,
    java.lang.String level3Obj,
    java.lang.String level4Obj,
    java.lang.String nativeEmsName,
    java.lang.String objectType,
    java.lang.String objectTypeQualifier,
    java.lang.String emsTime,
    java.lang.String neTime,
    java.lang.String isClearable,
    java.lang.String perceivedSeverity,
    java.lang.String layerRate,
    java.lang.String granularity,
    java.lang.String pmParameterName,
    java.lang.String pmLocation,
    java.lang.String thresholdType,
    java.lang.String value,
    java.lang.String unit,
    java.lang.String acknowledgeIndication)

Creates the n t_ tca.

Parameters:

notificationId - the notification id
ems - the ems
level2Obj - the level2 obj
level3Obj - the level3 obj
level4Obj - the level4 obj
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
perceivedSeverity - the perceived severity
layerRate - the layer rate
granularity - the granularity
pmParameterName - the pm parameter name
pmLocation - the pm location
thresholdType - the threshold type
value - the value
unit - the unit
acknowledgeIndication - the acknowledge indication

deleteTCAs
public void deleteTCAs(java.util.ArrayList list)

Delete tcas.

Parameters:

list - the list
**getAllActiveTCAs**

```java
public java.util.ArrayList getAllActiveTCAs()
```

Gets the all active tcas.

**Returns:**
the all active tcas

**getAllTCAParameters**

```java
public java.util.ArrayList getAllTCAParameters()
```

Gets the all tca parameters.

**Returns:**
the all tca parameters

**getTCA**

```java
public java.util.HashMap getTCA(java.lang.String notificationID)
```

Gets the tCA.

**Parameters:**
notificationID - the notification id

**Returns:**
the tCA

**getTCAs**

```java
public java.util.ArrayList getTCAs(java.lang.String[] serverityFilter,
                                 java.lang.String[] probCauseFilter)
```

Gets the tC as.

**Parameters:**
serverityFilter - the serverity filter
probCauseFilter - the prob cause filter

**Returns:**
the tC as
updateNTTCA

public java.lang.String updateNTTCA(java.lang.String notificationId,
java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String nativeEmsName,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String isClearable,
java.lang.String perceivedSeverity,
java.lang.String layerRate,
java.lang.String granularity,
java.lang.String pmParameterName,
java.lang.String pmLocation,
java.lang.String thresholdType,
java.lang.String value,
java.lang.String unit,
java.lang.String acknowledgeIndication)

Update nttca.

Parameters:

- notificationId - the notification id
- ems - the ems
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object
- nativeEmsName - the native ems name
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- perceivedSeverity - the perceived severity
- layerRate - the layer rate
- granularity - the granularity
- pmParameterName - the pm parameter name
- pmLocation - the pm location
- thresholdType - the threshold type
- value - the value
- unit - the unit
- acknowledgeIndication - the acknowledge indication

Returns:

- the string
Class ModelTI

java.lang.Object
  +--java.util.Observable
    +--ModelInit
    +--ModelHelper
    +--ModelEms
    +--ModelMe
    +--ModelPtp
    +--ModelCtp
    +--ModelMlsn
  +--com.ericsson.eos.model.ModelTl

Direct Known Subclasses:
  ModelTca

< Constructors > < Methods >

public class ModelTI
extends ModelMlsn

The Class ModelTI.

Constructors

ModelTI

public ModelTI()

Methods

deleteTL

public void deleteTL(java.lang.String ems,
                    java.lang.String tlName)

  Deletes the given topologicalLink.

  Parameters:
    ems - the ems
    tlName - the tl name
getAllTopologicalLinks

public java.util.ArrayList getAllTopologicalLinks()

Retrieves all TopologicalLinks.

Returns:
   an ArrayList of HashMaps containing attributes and values.

getTopologicalLink

public java.util.HashMap getTopologicalLink(int id)

gets the TopologicalLink with the specified objectId. used in MapView.

Parameters:
   id - the id

Returns:
   a HashMap containing attributes and values.
public void setTopologicalLink(java.lang.String emsName,
java.lang.String tlName,
java.lang.String userLabel,
java.lang.String nativeEmsName,
java.lang.String owner,
java.lang.String direction,
java.lang.String rate,
java.lang.String node1Ems,
java.lang.String node1Me,
java.lang.String node1PTP,
java.lang.String node1CTP,
java.lang.String node2Ems,
java.lang.String node2Me,
java.lang.String node2PTP,
java.lang.String node2CTP,
java.lang.String[] additionalInfo)

Creates a new TopologicalLink.

Parameters:

emsName - the ems name
tlName - the tl name
userLabel - the user label
nativeEmsName - the native ems name
owner - the owner
direction - the direction
rate - the rate
node1Ems - the node1 ems
node1Me - the node1 me
node1PTP - the node1 ptp
node1CTP - the node1 ctp
node2Ems - the node2 ems
node2Me - the node2 me
node2PTP - the node2 ptp
node2CTP - the node2 ctp
additionalInfo - the additional info
**updateTopologicalLink**

```java
public void updateTopologicalLink(java.lang.String emsName,
                                  java.lang.String oldTlName,
                                  java.lang.String newTlName,
                                  java.lang.String userLabel,
                                  java.lang.String nativeEmsName,
                                  java.lang.String owner,
                                  java.lang.String direction,
                                  java.lang.String rate,
                                  java.lang.String node1Ems,
                                  java.lang.String node1Me,
                                  java.lang.String node1PTP,
                                  java.lang.String node1CTP,
                                  java.lang.String node2Ems,
                                  java.lang.String node2Me,
                                  java.lang.String node2PTP,
                                  java.lang.String node2CTP,
                                  java.lang.String[] additionalInfo)
```

Changes the TopologicalLink.

**Parameters:**

- `emsName` - the ems name
- `oldTlName` - the old tl name
- `newTlName` - the new tl name
- `userLabel` - the user label
- `nativeEmsName` - the native ems name
- `owner` - the owner
- `direction` - the direction
- `rate` - the rate
- `node1Ems` - the node1 ems
- `node1Me` - the node1 me
- `node1PTP` - the node1 ptp
- `node1CTP` - the node1 ctp
- `node2Ems` - the node2 ems
- `node2Me` - the node2 me
- `node2PTP` - the node2 ptp
- `node2CTP` - the node2 ctp
- `additionalInfo` - the additional info

---

**com.ericsson.eos.model**

**Class MyTableModel**

```java
java.lang.Object
    |---javax.swing.table.AbstractTableModel
    |   |---javax.swing.table.DefaultTableModel
    |     |---com.ericsson.eos.model.MyTableModel
```

**All Implemented Interfaces:**

- java.io.Serializable, javax.swing.table.TableModel
public class MyTableModel
extends javax.swing.table.DefaultTableModel

The Class MyTableModel.

Constructors

MyTableModel

public MyTableModel()

Instantiates a new my table model.

MyTableModel

public MyTableModel(java.lang.Object[][] values,
                     java.lang.Object[] columns)

Instantiates a new my table model.

Parameters:
   values - the values
   columns - the columns

Methods

getColumnClass

public java.lang.Class getColumnClass(int columnIndex)

Overrides:
   getColumnClass in class javax.swing.table.AbstractTableModel

isCellEditable

public boolean isCellEditable(int rowIndex,
                               int columnIndex)

Overrides:
   isCellEditable in class javax.swing.table.DefaultTableModel
Package com.ericsson.eos.services

Class Summary

**NameService**

The Class NameService.

**NotificationService**

The Class NotificationService.

com.ericsson.eos.services

Class **NameService**

```plaintext
declares java.lang.Object
    |  +--- java.lang.Thread
    |     +--- com.ericsson.eos.services.NameService
```

All Implemented Interfaces:

```
java.lang.Runnable
```

< Constructors > < Methods >

**Constructors**

**NameService**

```plaintext
public NameService()
```

Instantiates a new name service.

**Methods**

**run**

```plaintext
public void run()
```

Overrides:

```
run in class java.lang.Thread
```
com.ericsson.eos.services

Class NotificationService

java.lang.Object
    +--java.lang.Thread
        +--com.ericsson.eos.services.NotificationService

All Implemented Interfaces:
    java.lang.Runnable

Constructors

public class NotificationService
extends java.lang.Thread

The Class NotificationService.

Constructors

NotificationService

public NotificationService()
    Instantiates a new notification service.

Methods

run

public void run()

    Overrides:
    run in class java.lang.Thread
Package com.ericsson.eos.view

Class Summary

AboutPanel
The Class AboutPanel.

EMSPanel
The Class EMSPanel.

MEPanel
The Class MEPanel.

MLSNPanel
The Class MLSNPanel.

MainView
The Class MainView.

ParseStatusPanel
The Class ParseStatusPanel.

StatusPanel
The Class StatusPanel includes the area in the bottom right corner of the application that displays the status of the database connection, the northbound connection and the connection with notification and name-service.

TPPanel
The Class PTPanel.

TPView
The Class TPPanel.

TopoLinkPanel
The Class TopoLinkPanel.

TreeNodeIconRenderer
The Class TreeNodeIconRenderer.

TreeView
This class generates the tree structure on the left side of the application window.

XkcdPanel
The Class XkcdPanel.
com.ericsson.eos.view

Class AboutPanel

java.lang.Object
    |-- java.awt.Component
        |-- java.awt.Container
            |-- javax.swing.JComponent
                |-- javax.swing.JPanel
                    |-- com.ericsson.eos.view.AboutPanel

All Implemented Interfaces:
    java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
    javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors >

public class AboutPanel
extends javax.swing.JPanel

The Class AboutPanel.

Constructors

AboutPanel

public AboutPanel()

    Instantiates a new about panel.
Class EMSPanel

java.lang.Object
   |---java.awt.Component
   |   |---java.awt.Container
   |   |   |---javax.swing.JComponent
   |   |   |   |---javax.swing.JPanel
   |   |---com.ericsson.eos.view.EMSPanel

All Implemented Interfaces:
   java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
   javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

Constructors

public class EMSPanel
extends javax.swing.JPanel

The Class EMSPanel.

Author:
   emikrie

EMSPanel

public EMSPanel(Model model,
                 java.awt.Dialog dialog)

   Creates new form MEPanel.

   Parameters:
      model - the model
dialog - the dialog
public EMSPanel(Model model,
    java.awt.Dialog dialog,
    java.lang.String ems,
    java.lang.String userLabel,
    java.lang.String nativeEmsName,
    java.lang.String owner,
    java.lang.String emsVersion,
    java.lang.String type)

Instantiates a new eMS panel.

Parameters:
    model - the model
    dialog - the dialog
    ems - the ems
    userLabel - the user label
    nativeEmsName - the native ems name
    owner - the owner
    emsVersion - the ems version
    type - the type

com.ericsson.eos.view

Class MEPanel

java.lang.Object
    +--java.awt.Component
        +--java.awt.Container
            +--javax.swing.JComponent
                +--javax.swing.JPanel
                    +--com.ericsson.eos.view.MEPanel

All Implemented Interfaces:
    java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
    javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors >

public class MEPanel
    extends javax.swing.JPanel

The Class MEPanel.

Constructors
MEPanel

public MEPanel(Model model,
        java.awt.Dialog dialog,
        java.lang.String ems,
        java.lang.String mlsn)

Creates new form MEPanel.

Parameters:
- model - the model
- dialog - the dialog
- ems - the ems
- mlsn - the mlsn

MEPanel

public MEPanel(Model model,
        java.awt.Dialog dialog,
        java.lang.String ems,
        java.lang.String meName,
        java.lang.String mlsnId,
        java.lang.String userLabel,
        java.lang.String location,
        java.lang.String version,
        java.lang.String productName,
        java.lang.String comState,
        java.lang.String nativeEMSName,
        boolean sync,
        java.lang.String owner)

Instantiates a mE panel.

Parameters:
- model - the model
- dialog - the dialog
- ems - the ems
- meName - the me name
- mlsnId - the mlsn id
- userLabel - the user label
- location - the location
- version - the version
- productName - the product name
- comState - the com state
- nativeEMSName - the native ems name
- sync - the sync
- owner - the owner
Class MLSNPanel

java.lang.Object
    ^--java.awt.Component
        ^--java.awt.Container
            ^--javax.swing.JComponent
                ^--javax.swing.JPanel
                    ^--com.ericsson.eos.view.MLSNPanel

All Implemented Interfaces:
    java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
    javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

constructors

public class MLSNPanel
extends javax.swing.JPanel

The Class MLSNPanel.

Author:
    emikrie

Constructors

MLSNPanel

public MLSNPanel(Model model,
                      java.awt.Dialog dialog,
                      java.lang.String ems)

    Creates new form MEPanel.

    Parameters:
        model - the model
        dialog - the dialog
        ems - the ems
**MLSNPanel**

```java
public class MLSNPanel(Model model,
java.awt.Dialog dialog,
java.lang.String ems,
java.lang.String mlsnName,
java.lang.String userLabel,
java.lang.String nativeEMSName,
java.lang.String type,
java.lang.String owner)

Instantiates a new mLSN panel.

**Parameters:**
- model - the model
- dialog - the dialog
- ems - the ems
- mlsnName - the mlsn name
- userLabel - the user label
- nativeEMSName - the native ems name
- type - the type
- owner - the owner
```

**com.ericsson.eos.view**

**Class MainView**

```java
java.lang.Object

++--java.awt.Component
  ++--java.awt.Container
    ++--java.awt.Window
      ++--java.awt.Frame
        ++--javax.swing.JFrame
          ++--com.ericsson.eos.view.MainView

All Implemented Interfaces:
  javax.accessibility.Accessible, javax.swing.RootPaneContainer,
  javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.WindowConstants
```

**Fields**

**public class MainView**

extends javax.swing.JFrame

implements java.util.Observer

The Class MainView.
**treeWidth**

```java
public static int treeWidth
    The tree width.
```

## Constructors

### MainView

```java
public MainView(Model m)
    Instantiates a new main view.
    Parameters:
    m - the m
```

## Methods

### getDebugFileMenu

```java
public javax.swing.JCheckBoxMenuItem getDebugFileMenu()
    Gets the debug file menu.
    Returns:
    the debug file menu
```

### getGroup

```java
public javax.swing.ButtonGroup getGroup()
    Gets the group.
    Returns:
    the group
```

### getMain

```java
public javax.swing.JPanel getMain()
    Gets the main.
    Returns:
    the main
```
getModel

public Model getModel()

    Gets the model.
    Returns:
    the model

getModelTreeSelection

public ModelSelection getModelTreeSelection()

    Gets the model tree selection.
    Returns:
    the model tree selection

defRTAMView

public RTAMView getRTAMView()

    Gets the rTAM view.
    Returns:
    the rTAM view

getSelectedInTree

public java.lang.String[] getSelectedInTree()

    Gets the selected in tree.
    Returns:
    the selected in tree

setSelectedInTree

public void setSelectedInTree(java.lang.String[] str)

    Sets the selected in tree.
    Parameters:
    str - the new selected in tree
switchView

public void switchView(java.lang.String str)

Switch view.

Parameters:

str - the str

update

public void update(java.util.Observable arg0,
                     java.lang.Object arg1)

com.ericsson.eos.view

Class ParseStatusPanel

does not implement Serializable

java.lang.Object
   +--java.awt.Component
      +--java.awt.Container
         +--javax.swing.JComponent
            +--javax.swing.JPanel
               +--com.ericsson.eos.view.ParseStatusPanel

All Implemented Interfaces:
   javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class ParseStatusPanel
   extends javax.swing.JPanel
   implements java.util.Observer

The Class ParseStatusPanel.

Constructors
**ParseStatusPanel**

```java
public ParseStatusPanel(Model model, java.awt.Dialog d)
```

Creates new form ParseStatusPanel.

**Parameters:**
- `model` - the model
- `d` - the d

### Methods

**print**

```java
public void print()
```

For each different type of element, this function prints the amount of elements that were imported as well as the amount that failed on the top of the panel.

**update**

```java
public void update(java.util.Observable arg0, java.lang.Object mess)
```

### com.ericsson.eos.view

**Class StatusPanel**

```java
java.lang.Object
  |--java.awt.Component
    |--java.awt.Container
      |--javax.swing.JComponent
        |--javax.swing.JPanel
          |--com.ericsson.eos.view.StatusPanel
```

**All Implemented Interfaces:**

< Constructors > < Methods >
The Class StatusPanel includes the area in the bottom right corner of the application that displays the status of the database connection, the northbound connection and the connection with notification and name-service.

**Constructors**

**StatusPanel**

```java
public StatusPanel(Model model)
```

Instantiates a new status panel.

**Parameters:**

model - the model

**Methods**

**repaintStatus**

```java
public void repaintStatus()
```

Repaint status.

**update**

```java
public void update(java.util.Observable arg0, java.lang.Object arg1)
```

**com.ericsson.eos.view**

**Class TPPanel**

```java
java.lang.Object
    |   +--java.awt.Component
    |       |   +--java.awt.Container
    |       |       |   +--javax.swing.JComponent
    |       |       |       |   +--javax.swing.JPanel
    |       |       |       |       |   +--com.ericsson.eos.view.TPPanel
```

**All Implemented Interfaces:**


< Constructors >
public class TPanel
extends javax.swing.JPanel

The Class TPanel.

Author:
    emikrie

Constructors

TPanel

public TPanel(Model model,
                java.awt.Dialog dialog,
                java.lang.String ems,
                java.lang.String me)

  Instantiates a new TPanel.

  Parameters:
      model - the model
      dialog - the dialog
      ems - the ems
      me - the me

TPanel

public TPanel(Model model,
                java.awt.Dialog dialog,
                java.lang.String ems,
                java.lang.String me,
                java.lang.String ptp)

  Instantiates a new TPanel.

  Parameters:
      model - the model
      dialog - the dialog
      ems - the ems
      me - the me
      ptp - the ptp
Instantiates a new pTP panel.

**Parameters:**

- `model`: the model
- `dialog`: the dialog
- `ems`: the ems
- `me`: the me
- `ptp`: the ptp
- `ctp`: the ctp
- `userLabel`: the user label
- `nativeEmsName`: the native ems name
- `owner`: the owner
- `ingressName`: the ingress name
- `egressName`: the egress name
- `type`: the type
- `connectionState`: the connection state
- `mappingMode`: the mapping mode
- `directionality`: the directionality
- `protectionAssosiation`: the protection association
- `edgePoint`: the edge point
The Class TPPanel.

Constructors

TPView

public TPView(Model model, MainView mv, ModelSelection mts)

Creates new form TPView.

Parameters:

model - the model
mv - the mv
mts - the mts

Methods

getCTPMenu

public javax.swing.JPopupMenu getCTPMenu()

Gets the cTP menu.

Returns:

the cTP menu
getCTPTable
public javax.swing.JTable getCTPTable()

Gets the cTP table.

Returns:
the cTP table

getMCS
public ModelSelection getMCS()

Gets the mCS.

Returns:
the mCS

getMPS
public ModelSelection getMPS()

Gets the mPS.

Returns:
the mPS

getMTS
public ModelSelection getMTS()

Gets the mTS.

Returns:
the mTS

getMenu
public javax.swing.JPopupMenu getMenu()

Gets the menu.

Returns:
the menu
**getPTPTable**

```java
getPTPTable()
```

Gets the pTP table.

**Returns:**
the pTP table

**setCtpModel**

```java
setCtpModel(javax.swing.table.DefaultTableModel ctp)
```

Sets the ctp model.

**Parameters:**
ctp - the new ctp model

**setPtpModel**

```java
setPtpModel(javax.swing.table.DefaultTableModel ptp)
```

Sets the ptp model.

**Parameters:**
ptp - the new ptp model

**update**

```java
update(java.util.Observable arg0, java.lang.Object arg1)
```
Class TopoLinkPanel

java.lang.Object
  |---java.awt.Component
  |   |---java.awt.Container
  |     |---javax.swing.JComponent
  |        |---javax.swing.JPanel
  |           |---com.ericsson.eos.view.TopoLinkPanel

All Implemented Interfaces:
  java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
  javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class TopoLinkPanel
extends javax.swing.JPanel

The Class TopoLinkPanel.

Constructors

public TopoLinkPanel(Model model,
                      java.awt.Dialog dialog,
                      java.lang.String ems,
                      java.lang.String nodeAMe,
                      java.lang.String nodeZMe)

Instantiates a new topo link panel.

Parameters:
    model - the model
dialog - the dialog
esms - the ems
nodeAMe - the node a me
nodeZMe - the node z me
TopoLinkPanel

public TopoLinkPanel(Model model,
                java.awt.Dialog dialog,
                java.lang.String ems,
                java.lang.String topologicalLink,
                java.lang.String userLabel,
                java.lang.String nativeEmsName,
                java.lang.String owner,
                java.lang.String direction,
                java.lang.String layerRate,
                java.lang.String[] startPoint,
                java.lang.String[] stopPoint)

Instantiates a new topo link panel.

Parameters:

- model - the model
- dialog - the dialog
- ems - the ems
- topologicalLink - the topological link
- userLabel - the user label
- nativeEmsName - the native ems name
- owner - the owner
- direction - the direction
- layerRate - the layer rate
- startPoint - the start point
- stopPoint - the stop point

Methods

fixLR

public void fixLR()

Fix lr.

getCTPs

public java.lang.String[] getCTPs(java.lang.String ems,
                java.lang.String me,
                java.lang.String ptp)

Gets the cT ps.

Parameters:

- ems - the ems
- me - the me
- ptp - the ptp

Returns:

- the cT ps
getPTPs

public java.lang.String[] getPTPs(java.lang.String ems, java.lang.String me)

Gets the pT ps.

**Parameters:**

ems - the ems
me - the me

**Returns:**

the pT ps

com.ericsson.eos.view

**Class TreeNodeIconRenderer**

java.lang.Object
  +--java.awt.Component
    +--java.awt.Container
      +--javax.swing.JComponent
        +--javax.swing.JLabel
          +--javax.swing.tree.DefaultTreeCellRenderer
            +--com.ericsson.eos.view.TreeNodeIconRenderer

**All Implemented Interfaces:**
java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.SwingConstants,
javax.swing.TransferHandler.HasGetTransferHandler, javax.swing.tree.TreeCellRenderer

< Constructors > < Methods >

public class TreeNodeIconRenderer
extends javax.swing.tree.DefaultTreeCellRenderer

The Class TreeNodeIconRenderer.

**Constructors**

**TreeNodeIconRenderer**

public TreeNodeIconRenderer(Model model)

Instantiates a new tree node icon renderer.

**Parameters:**

model - the model
Methods

getTreeCellRendererComponent

```java
public java.awt.Component getTreeCellRendererComponent(javax.swing.JTree tree,
java.lang.Object value,
boolean sel,
boolean expanded,
boolean leaf,
int row,
boolean hasFocus)
```

Overrides:

getTreeCellRendererComponent in class javax.swing.tree.DefaultTreeCellRenderer

com.ericsson.eos.view

**Class TreeView**

```
java.lang.Object
     |---java.awt.Component
      |---java.awt.Container
      |     |---javax.swing.JComponent
      |     |     |---javax.swing.JPanel
      |     +---com.ericsson.eos.view.TreeView
```

All Implemented Interfaces:

javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class TreeView
extends javax.swing.JPanel
implements java.util.Observer

This class generates the tree structure on the left side of the application window.

Constructors
**TreeView**

```java
public TreeView(Model m,
                MainController mc,
                MainView mv)
```

Instantiates a new tree view.

**Parameters:**

- m - the m
- mc - the mc
- mv - the mv

**Methods**

**getMenuEMS**

```java
public javax.swing.JPopupMenu getMenuEMS()
```

Gets the menu ems.

**Returns:**

- the menu ems

**getMenuME**

```java
public javax.swing.JPopupMenu getMenuME()
```

Gets the menu me.

**Returns:**

- the menu me

**getMenuML**

```java
public javax.swing.JPopupMenu getMenuML()
```

Gets the menu ml.

**Returns:**

- the menu ml

**getRootNode**

```java
public javax.swing.tree.DefaultMutableTreeNode getRootNode()
```

Gets the root node.

**Returns:**

- the root node
**getSelectedNodeCount**

```java
public int getSelectedNodeCount()

    Gets the selected node count.
    Returns:
    the selected node count
```

**getSelectedNodes**

```java
public java.util.ArrayList getSelectedNodes()

    Gets the selected nodes.
    Returns:
    the selected nodes
```

**getTree**

```java
public javax.swing.JTree getTree()

    Gets the tree.
    Returns:
    the tree
```

**getTreeModel**

```java
public javax.swing.tree.DefaultTreeModel getTreeModel()

    Gets the tree model.
    Returns:
    the tree model
```

**getTreeView**

```java
public javax.swing.JScrollPane getTreeView()

    Gets the tree view.
    Returns:
    the tree view
```
**setHeartBeatMenu**

public void `setHeartBeatMenu(boolean bool)`

Sets the heart beat menu.

**Parameters:**

bool - the new heart beat menu

**update**

public void `update(java.util.Observable arg0, java.lang.Object arg1)`

---

**com.ericsson.eos.view**

**Class XkcdPanel**

`java.lang.Object`

+--`com.ericsson.eos.view.XkcdPanel`

**All Implemented Interfaces:**

`java.awt.event.MouseListener`

< Constructors > < Methods >

public class `XkcdPanel`

extends `java.lang.Object`

implements `java.awt.event.MouseListener`

The Class XkcdPanel.

**Author:**

`emikrie`

---

**Constructors**

**XkcdPanel**

public `XkcdPanel()`

Instantiates a new xkcd panel.

---

**Methods**
```java
public void mouseClicked(MouseEvent arg0)

public void mouseEntered(MouseEvent arg0)

public void mouseExited(MouseEvent arg0)

public void mousePressed(MouseEvent arg0)

public void mouseReleased(MouseEvent arg0)
```
Package com.ericsson.eos.view.alarm

Class Summary

**AlarmAlarmPanel**
This class.

**AlarmTCAPanel**
The Class AlarmTCAPanel.

**NTAlarmView**
This class creates the panel for creating and configuring an nt_alarm.

**NTTCAView**
The Class NTTCAView.

---

com.ericsson.eos.view.alarm

Class **AlarmAlarmPanel**

```java
java.lang.Object
   |--java.awt.Component
      |--java.awt.Container
         |--javax.swing.JComponent
            |--javax.swing.JPanel
                |--com.ericsson.eos.view.alarm.AlarmAlarmPanel
```

All Implemented Interfaces:
java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class **AlarmAlarmPanel**
extends javax.swing.JPanel

This class.

 Constructors

**AlarmAlarmPanel**

```java
public AlarmAlarmPanel()
```

Creates new form AlarmAlarmPanel.
public AlarmAlarmPanel(Model model,
    java.lang.String ems,
    java.lang.String level2,
    java.lang.String level3,
    java.lang.String level4,
    java.lang.String type)

Instantiates a new alarm alarm panel.

Parameters:
    model - the model
    ems - the ems
    level2 - the level2
    level3 - the level3
    level4 - the level4
    type - the type
public AlarmAlarmPanel(Model model,
        java.lang.String notificationID,
        java.lang.String ems,
        java.lang.String level2,
        java.lang.String level3,
        java.lang.String level4,
        java.lang.String nativeEMSName,
        java.lang.String nativeProbCause,
        java.lang.String objectType,
        java.lang.String objectTypeQualifier,
        java.lang.String emsTime,
        java.lang.String neTime,
        java.lang.String isClearable,
        java.lang.String layerRate,
        java.lang.String probableCause,
        java.lang.String probableCauseQualifier,
        java.lang.String perceivedSeverity,
        java.lang.String serviceAffecting,
        java.lang.String additionalText,
        java.lang.String[] affectedTPList,
        java.lang.String x733EventType,
        java.lang.String x733BackedUpStatus,
        java.lang.String x733BackupObject,
        java.lang.String x733TrendIndication,
        java.lang.String[] x733SpecificProblems,
        java.lang.String[] x733ProposedRepairActions,
        java.lang.String[] x733CorrelatedNotifications,
        java.lang.String[] x733MonitoredAttributes,
        java.lang.String[] x733AdditionalInfo,
        java.lang.String rcaiIndicator,
        java.lang.String acknowledgeIndication)

Instantiates a new alarm alarm panel.

**Parameters:**

- model - the model
- notificationID - the notification id
- ems - the ems
- level2 - the level2
- level3 - the level3
- level4 - the level4
- nativeEMSName - the native ems name
- nativeProbCause - the native prob cause
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- layerRate - the layer rate
- probableCause - the probable cause
- probableCauseQualifier - the probable cause qualifier
- perceivedSeverity - the perceived severity
- serviceAffecting - the service affecting
- additionalText - the additional text
- affectedTPList - the affected tp list
- x733EventType - the x733 event type
- x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733TrendIndication - the x733 trend indication
x733SpecificProblems - the x733 specific problems
x733ProposedRepairActions - the x733 proposed repair actions
x733CorrelatedNotifications - the x733 correlated notifications
x733MonitoredAttributes - the x733 monitored attributes
x733AdditionalInfo - the x733 additional info
rcaiIndicator - the rcai indicator
acknowledgeIndication - the acknowledge indication

Methods

createAlarm

public boolean createAlarm()

Creates the alarm.

Returns:
true, if successful

fixLR

public void fixLR()

Fix lr.

reset

public void reset()

Reset.

setModel

public void setModel()

Sets the model.
public void setTemplate(java.lang.String type,
    java.util.ArrayList alarmDef,
    java.util.Date dateTime)

Sets the template.

Parameters:
    type - the type
    alarmDef - the alarm def
    dateTime - the date time
public AlarmTCAPanel(Model model,
                      java.lang.String ems,
                      java.lang.String level2,
                      java.lang.String level3,
                      java.lang.String level4,
                      java.lang.String type)

Instantiates a new alarm tca panel.

Parameters:

  model - the model
  ems - the ems
  level2 - the level2
  level3 - the level3
  level4 - the level4
  type - the type

Instantiates a new alarm tca panel.

Parameters:

model - the model
notificationId - the notification id
ems - the ems
level2 - the level2
level3 - the level3
level4 - the level4
nativeEmsName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
layerRate - the layer rate
perceivedSeverity - the perceived severity
granularity - the granularity
pmParameterName - the pm parameter name
pmLocation - the pm location
thresholdType - the threshold type
value - the value
unit - the unit
acknowledgedIndication - the acknowledged indication

Methods

createAlarm

public void createAlarm()

Creates the alarm.
public void fixLR()
    Fix lr.

public void reset()
    Reset.

public void setModel()
    Sets the model.

com.ericsson.eos.view.alarm

Class NTAlarmView


This class creates the panel for creating and configuring an nt_alarm.
public NTAlarmView(Model model,
        java.awt.Dialog d,
        java.lang.String ems,
        java.lang.String level2,
        java.lang.String level3,
        java.lang.String level4,
        java.lang.String type)

Instantiates a new nT alarm view.

Parameters:

model - the model
d - the d
ems - the ems
level2 - the level2
level3 - the level3
level4 - the level4
type - the type
**NTAlarmView**

public NTAlarmView(Model model,
        java.awt.Dialog d,
        java.lang.String notificationID,
        java.lang.String ems,
        java.lang.String level2,
        java.lang.String level3,
        java.lang.String level4,
        java.lang.String nativeEMSName,
        java.lang.String nativeProbCause,
        java.lang.String objectType,
        java.lang.String objectTypeQualifier,
        java.lang.String emsTime,
        java.lang.String neTime,
        java.lang.String isClearable,
        java.lang.String layerRate,
        java.lang.String probableCause,
        java.lang.String probableCauseQualifier,
        java.lang.String perceivedSeverity,
        java.lang.String serviceAffecting,
        java.lang.String additionalText,
        java.lang.String[] affectedTPList,
        java.lang.String x733EventType,
        java.lang.String x733BackedUpStatus,
        java.lang.String x733BackupObject,
        java.lang.String x733TrendIndication,
        java.lang.String[] x733SpecificProblems,
        java.lang.String[] x733ProposedRepairActions,
        java.lang.String[] x733CorrelatedNotifications,
        java.lang.String[] x733MonitoredAttributes,
        java.lang.String[] x733AdditionalInfo,
        java.lang.String rcaiIndicator,
        java.lang.String acknowledgeIndication)

Instantiates a new nT alarm view.

**Parameters:**

- model - the model
- d - the d
- notificationID - the notification id
- ems - the ems
- level2 - the level2
- level3 - the level3
- level4 - the level4
- nativeEMSName - the native ems name
- nativeProbCause - the native prob cause
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- isClearable - the is clearable
- layerRate - the layer rate
- probableCause - the probable cause
- probableCauseQualifier - the probable cause qualifier
- perceivedSeverity - the perceived severity
- serviceAffecting - the service affecting
- additionalText - the additional text
- affectedTPList - the affected tp list
- x733EventType - the x733 event type
x733BackedUpStatus - the x733 backed up status
x733BackupObject - the x733 backup object
x733TrendIndication - the x733 trend indication
x733SpecificProblems - the x733 specific problems
x733ProposedRepairActions - the x733 proposed repair actions
x733CorrelatedNotifications - the x733 correlated notifications
x733MonitoredAttributes - the x733 monitored attributes
x733AddtionalInfo - the x733 additional info
rcaiIndicator - the rcai indicator
acknowledgeIndication - the acknowledge indication

Methods

setModel

public javax.swing.DefaultComboBoxModel setModel()

Sets the model.

Returns:
the default combo box model

com.ericsson.eos.view.alarm

Class NTTCAView

java.lang.Object
  +--java.awt.Component
    +--java.awt.Container
      +--javax.swing.JComponent
        +--javax.swing.JPanel
          +--com.ericsson.eos.view.alarm.NTTCAView

All Implemented Interfaces:
java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class NTTCAView
extends javax.swing.JPanel

The Class NTTCAView.

Author:
emikrie

Constructors
NTTCAView

public NTTCAView(Model model,
                   java.awt.Dialog d,
                   java.lang.String ems,
                   java.lang.String level2,
                   java.lang.String level3,
                   java.lang.String level4,
                   java.lang.String type)

Creates new form AlarmPanel.

Parameters:

   model - the model
   d - the d
   ems - the ems
   level2 - the level2
   level3 - the level3
   level4 - the level4
   type - the type
public NTTCAView(Model model,
        java.awt.Dialog d,
        java.lang.String notificationID,
        java.lang.String ems,
        java.lang.String level2,
        java.lang.String level3,
        java.lang.String level4,
        java.lang.String nativeEMSName,
        java.lang.String objectType,
        java.lang.String objectTypeQualifier,
        java.lang.String emsTime,
        java.lang.String neTime,
        java.lang.String isClearable,
        java.lang.String layerRate,
        java.lang.String perceivedSeverity,
        java.lang.String granularity,
        java.lang.String pmParameterName,
        java.lang.String pmLocation,
        java.lang.String pmThresholdType,
        java.lang.String pmValue,
        java.lang.String pmUnit,
        java.lang.String acknowledgeIndication)

Instantiates a new nTTCA view.

Parameters:

model - the model
d - the d
notificationID - the notification id
ems - the ems
level2 - the level2
level3 - the level3
level4 - the level4
nativeEMSName - the native ems name
objectType - the object type
objectTypeQualifier - the object type qualifier
emsTime - the ems time
neTime - the ne time
isClearable - the is clearable
layerRate - the layer rate
perceivedSeverity - the perceived severity
granularity - the granularity
pmParameterName - the pm parameter name
pmLocation - the pm location
pmThresholdType - the pm threshold type
pmValue - the pm value
pmUnit - the pm unit
acknowledgeIndication - the acknowledge indication
**setModel**

```java
public javax.swing.DefaultComboBoxModel ``` **setModel** `()`

Sets the model.

**Returns:**

the default combo box model
Package com.ericsson.eos.view.rtam

Class Summary

**CustomTableCellRenderer**
The Class CustomTableCellRenderer.

**RTAMAlarm**
The Class RTAMAlarm.

**RTAMFilter**
The Class RTAMFilter.

**RTAMTCA**
The Class RTAMTCA.

**RTAMView**
The Class RTAMView.

---

com.ericsson.eos.view.rtam

Class CustomTableCellRenderer

does not implement any interfaces.

public class CustomTableCellRenderer
extends javax.swing.table.DefaultTableCellRenderer

The Class CustomTableCellRenderer.

---

Constructors

public class CustomTableCellRenderer
extends javax.swing.table.DefaultTableCellRenderer

The Class CustomTableCellRenderer.
CustomTableCellRenderer

public CustomTableCellRenderer(Model model)

Instantiates a new custom table cell renderer.

Parameters:
model - the model

Methods

getTableCellRendererComponent

public java.awt.Component getTableCellRendererComponent(javax.swing.JTable table,
java.lang.Object obj,
boolean isSelected,
boolean hasFocus,
int row,
int column)

Overrides:
getTableCellRendererComponent in class javax.swing.table.DefaultTableCellRenderer

com.ericsson.eos.view.rtam

Class RTAMAlarm

java.lang.Object
  |--java.awt.Component
  |  |--java.awt.Container
  |  |  |--javax.swing.JComponent
  |  |  |  |--javax.swing.JPanel
  |  |  |  |  |--com.ericsson.eos.view.rtam.RTAMAlarm

All Implemented Interfaces:
javax.accessibility.Accessible, javax.swing.TransferHandler HASGetTransferHandler

< Constructors > < Methods >

public class RTAMAlarm
extends javax.swing.JPanel
implements java.util.Observer

The Class RTAMAlarm.
Constructors

RTAMAlarm

public RTAMAlarm(Model m,
RTAMView rtam)

Instantiates a new rTAM alarm.

Parameters:

m - the m
rtam - the rtam

Methods

getColumnNames

public java.lang.String[] getColumnNames()

Gets the column names.

Returns:

the column names

getMenu

public javax.swing.JPopupMenu getMenu()

Gets the menu.

Returns:

the menu

getModel

public Model getModel()

Gets the model.

Returns:

the model
**getTable**

```java
public javax.swing.JTable getTable()
```

Gets the table.

**Returns:**

the table

---

**getViewColumns**

```java
public java.lang.String[] getViewColumns()
```

Gets the view columns.

**Returns:**

the view columns

---

**newFilter**

```java
public void newFilter(java.util.HashMap hm)
```

New filter.

**Parameters:**

hm - the hm

---

**setColumnNames**

```java
public void setColumnNames(java.lang.String[] columnNames)
```

Sets the column names.

**Parameters:**

columnNames - the new column names

---

**update**

```java
public void update(java.util.Observable o,
java.lang.Object arg)
```
Class RTAMFilter

java.lang.Object
  |-- java.awt.Component
    |-- java.awt.Container
      |-- javax.swing.JComponent
        |-- javax.swing.JPanel
          |-- com.ericsson.eos.view.rtam.RTAMFilter

All Implemented Interfaces:
  java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
  javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

Fields

acknowledgeIndication

public javax.swing.JComboBox acknowledgeIndication
The acknowledge indication.

serviceAffecting

public javax.swing.JComboBox serviceAffecting
The service affecting.

Constructors

RTAMFilter

public RTAMFilter(RTAMView rv)
Instantiates a new rTAM filter.

Parameters:
  rv - the rv

Methods
getFilterData

public java.util.HashMap getFilterData()

Gets the filter data.

Returns:
the filter data

getMode

public java.lang.String getMode()

Gets the mode.

Returns:
the mode

com.ericsson.eos.view.rtam

Class RTAMTCA

java.lang.Object
    +-- java.awt.Component
        +-- java.awt.Container
            +-- javax.swing.JComponent
                +-- javax.swing.JPanel
                    +-- com.ericsson.eos.view.rtam.RTAMTCA

All Implemented Interfaces:
    javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class RTAMTCA
extends javax.swing.JPanel
implements java.util.Observer

The Class RTAMTCA.

Constructors
RTAMTCA

public RTAMTCA(Model m, RTAMView rtam)

Instantiates a new RTAMTCA.

Parameters:
  m - the m
  rtam - the rtam

Methods

cgetColumnNames

public java.lang.String[] getColumnNames()

Gets the column names.

Returns:
  the column names

cgetMenu

public javax.swing.JPopupMenu getMenu()

Gets the menu.

Returns:
  the menu

cgetModel

public Model getModel()

Gets the model.

Returns:
  the model

cgetTable

public javax.swing.JTable getTable()

Gets the table.

Returns:
  the table
getViewColumns

public java.lang.String[] getViewColumns()

Gets the view columns.

Returns:

the view columns

newFilter

public void newFilter(java.util.HashMap hm)

New filter.

Parameters:

hm - the hm

setColumnNames

public void setColumnNames(java.lang.String[] columnNames)

Sets the column names.

Parameters:

columnNames - the new column names

update

public void update(java.util.Observable o,
java.lang.Object arg)
Class RTAMView

java.lang.Object
    |---java.awt.Component
        |---java.awt.Container
            |---javax.swing.JComponent
                |---javax.swing.JPanel
                    +--com.ericsson.eos.view.rtam.RTAMView

All Implemented Interfaces:
    java.awt.MenuContainer, java.awt.image.ImageObserver, java.io.Serializable,
    javax.accessibility.Accessible, javax.swing.TransferHandler.HasGetTransferHandler

< Constructors > < Methods >

public class RTAMView
extends javax.swing.JPanel

The Class RTAMView.

Constructors

RTAMView

public RTAMView(Model model,
                MainView mv)

    Instantiates a new rTAM view.
    Parameters:
        model - the model
        mv - the mv

Methods

getController

public RTAMController getController()

    Gets the controller.
    Returns:
        the controller
**getModel**

public Model getModel()

    Gets the model.

    Returns:
    the model

---

**getRTAMAlarm**

public RTAMAlarm getRTAMAlarm()

    Gets the rTAM alarm.

    Returns:
    the rTAM alarm

---

**getRTAMTCA**

public RTAMTCA getRTAMTCA()

    Gets the rTAMTCA.

    Returns:
    the rTAMTCA

---

**getSelectedMenu**

public javax.swing.JPopupMenu getSelectedMenu()

    Gets the selected menu.

    Returns:
    the selected menu

---

**getSelectedTable**

public javax.swing.JTable getSelectedTable()

    Gets the selected table.

    Returns:
    the selected JTable
init

public void init()

  Inits the.

newFilter

public void newFilter()

  New filter.

switchView

public void switchView(java.lang.String str)

  Switch view.
  Parameters:
  str - the str
Class Summary

**AlarmCreator**

The Class AlarmCreator.

**NameExtractor**

The Class NameExtractor.

---

com.ericsson.helper

**Class AlarmCreator**

```
java.lang.Object
   +--com.ericsson.helper.AlarmCreator

All Implemented Interfaces:
   java.io.Serializable
```

< Constructors > < Methods >

public class **AlarmCreator**
extends java.lang.Object
implements java.io.Serializable

The Class AlarmCreator.

---

**Constructors**

**AlarmCreator**

```java
public AlarmCreator()
```

---

**Methods**
createNT_AlarmEvent

public static org.omg.CosNotification.StructuredEvent createNT_AlarmEvent(java.util.HashMap al,
NBIImp nbi)

Creates the n t_ alarm event.

Parameters:
   al - the al
   nbi - the nbi

Returns:
   the structured event

---

createNT_Attribute_Value_Change

public static org.omg.CosNotification.StructuredEvent createNT_Attribute_Value_Change(java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String notificationId,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String edgePointRelated,
java.lang.String[] attributList,
NBIImp nbi)

Creates the n t_ attribute_ value_ change.

Parameters:
   ems - the ems
   level2Object - the level2 object
   level3Object - the level3 object
   level4Object - the level4 object
   notificationId - the notification id
   objectType - the object type
   objectTypeQualifier - the object type qualifier
   emsTime - the ems time
   neTime - the ne time
   edgePointRelated - the edge point related
   attributList - the attribut list
   nbi - the nbi

Returns:
   the structured event
createNT_HeartBeatEvent

```java
public static org.omg.CosNotification.StructuredEvent createNT_HeartBeatEvent(java.lang.String ems,
java.lang.String me,
java.lang.String notificationId,
java.lang.String emsTime,
NBIImp nbi)
```

Creates the n t_ heart beat event.

**Parameters:**
- ems - the ems
- me - the me
- notificationId - the notification id
- emsTime - the ems time
- nbi - the nbi

**Returns:**
- the structured event

createNT_State_Change

```java
public static org.omg.CosNotification.StructuredEvent createNT_State_Change(java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String notificationId,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String edgePointRelated,
java.lang.String[] attributList,
NBIImp nbi)
```

Creates the n t_ state_ change.

**Parameters:**
- ems - the ems
- level2Object - the level2 object
- level3Object - the level3 object
- level4Object - the level4 object
- notificationId - the notification id
- objectType - the object type
- objectTypeQualifier - the object type qualifier
- emsTime - the ems time
- neTime - the ne time
- edgePointRelated - the edge point related
- attributList - the attribut list
- nbi - the nbi

**Returns:**
- the structured event
createNT_TCAEvent

public static org.omg.CosNotification.StructuredEvent createNT_TCAEvent(java.util.HashMap al, NBIImp nbi)

    Creates the n t_ tca event.

    Parameters:
    al - the al
    nbi - the nbi

    Returns:
    the structured event

printProperty

public static void printProperty(org.omg.CosNotification.Property[] props)

    Prints the property.

    Parameters:
    props - the props

com.ericsson.helper

Class NameExtractor

java.lang.Object
    |--com.ericsson.helper.NameExtractor

All Implemented Interfaces:
    java.io.Serializable

< Methods >

public class NameExtractor
extends java.lang.Object
implements java.io.Serializable

    The Class NameExtractor.

Methods
getObject

public static java.lang.String[]
getObject(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName)

   Gets the object name in a list.

   Parameters:
      objectName - the object name

   Returns:
      the object
The Class NBIImp.

The event iterators.

The managed element iterators.

The naming attributes iterators.
subnetworkIterators

public java.util.ArrayList subnetworkIterators
  The subnetwork iterators.

terminationPointIterators

public java.util.ArrayList terminationPointIterators
  The termination point iterators.

Constructors

NBIImp

public NBIImp()
  Instantiates a new nBI imp.

Methods

acknowledgeAlarms

public java.util.ArrayList acknowledgeAlarms(java.util.ArrayList alarms,
                                             java.util.ArrayList addInfo)

debug

public void debug(java.lang.String str,
                  java.lang.Object o,
                  java.util.logging.Level level)

getActive

public boolean getActive()

getAdditionalInfo

public java.util.ArrayList getAdditionalInfo(int id)
getAlarms
public java.util.ArrayList getAlarms(java.lang.String[] severity,
java.lang.String[] probableCause)

getAllAlarms
public java.lang.String[] getAllAlarms()

getAllManagedElements
public java.util.ArrayList getAllManagedElements()

getAllManagedElements
public java.util.ArrayList getAllManagedElements(java.lang.String ems,
java.lang.String subnetwork)

getAllNodes
public java.lang.String[] getAllNodes()

getAllTL
public java.lang.String[] getAllTL()

getAllTopLevelSubnetworkLayerRates
public java.util.ArrayList getAllTopLevelSubnetworkLayerRates(int ems,
java.lang.String subnet)

getAllTopLevelSubnetworks
public java.util.ArrayList getAllTopLevelSubnetworks()
getCTP

```java
public java.util.HashMap getCTP(java.lang.String ems,
                                 java.lang.String me,
                                 java.lang.String ptp,
                                 java.lang.String ctp)
```

getChannel

```java
public org.omg.CosNotifyChannelAdmin.EventChannel getChannel()

    Gets the channel.
    
    Returns:
    the channel
```

ggetClient

```java
public org.tmforum.mtnm.nmsSession.NmsSession_I getClient()

    Gets the client.
    
    Returns:
    the client
```

ggetContainedCurrentTPNames

```java
public java.lang.String[] getContainedCurrentTPNames(java.lang.String ems,
                                                      java.lang.String me,
                                                      java.lang.String tp,
                                                      short[] layerRate)
```

ggetContainedCurrentTPs

```java
public java.util.ArrayList getContainedCurrentTPs(java.lang.String ems,
                                                java.lang.String me,
                                                java.lang.String tp,
                                                short[] layerRate)
```
getContainedInUseTPNames
public java.lang.String[] getContainedInUseTPNames(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

getContainedInUseTPs
public java.util.ArrayList getContainedInUseTPs(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

getContainedPotentialTPNames
public java.lang.String[] getContainedPotentialTPNames(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

getContainedPotentialTPs
public java.util.ArrayList getContainedPotentialTPs(java.lang.String ems,
java.lang.String me,
java.lang.String tp,
short[] layerRate)

getEms
public java.util.HashMap getEms()

getInterfaceName
public java.lang.String getInterfaceName()
getLayerParameters
public java.util.ArrayList getLayerParameters(int objectID)

getManagedElement
public java.util.HashMap getManagedElement(java.lang.String ems,
java.lang.String me)

getNodes
public java.lang.String[] getNodes(java.lang.String node)

getObjectName
public java.util.HashMap getObjectName(int objectId)

getOrb
public org.omg.CORBA.ORB getOrb()
    Gets the orb.
    Returns:
    the orb

getPOA
public org.omg.PortableServer.POA getPOA()
    Gets the pOA.
    Returns:
    the pOA

getPTP
public java.util.HashMap getPTP(java.lang.String ems,
java.lang.String me,
java.lang.String ptp)
### getPTPs

```java
public java.util.ArrayList getPTPs(java.lang.String ems,
                                          java.lang.String me)
```

### getSubnetwork

```java
public java.util.HashMap getSubnetwork(java.lang.String ems,
                                          java.lang.String subnetwork)
```

### getSubnodes

```java
public java.lang.String[] getSubnodes(java.lang.String node)
```

### getSupportedRates

```java
public java.util.HashMap getSupportedRates(int id)
```

### getTL

```java
public java.lang.String[] getTL(java.lang.String node)
```

### getTopNode

```java
public java.lang.String getTopNode()
```

### getX733AdditionalInfo

```java
public java.util.ArrayList getX733AdditionalInfo(java.lang.String id)
```

### getX733MonitoredAttribute

```java
public java.util.ArrayList getX733MonitoredAttribute(java.lang.String id)
```
notificate
public void notificate(org.omg.CosNotification.StructuredEvent se)

run
public void run()

Overrides:
run in class java.lang.Thread

sendAlarm
public void sendAlarm(java.util.HashMap alarm)

sendAttributeChanged
public void sendAttributeChanged(java.lang.String ems,
java.lang.String level2Object,
java.lang.String level3Object,
java.lang.String level4Object,
java.lang.String notificationId,
java.lang.String objectType,
java.lang.String objectTypeQualifier,
java.lang.String emsTime,
java.lang.String neTime,
java.lang.String edgePointRelated,
java.lang.String[] attributList)

sendHeartBeat
public void sendHeartBeat(java.lang.String ems,
java.lang.String me,
java.lang.String notificationId,
java.lang.String emsTime)
sendStateChanged

public void sendStateChanged(java.lang.String emsName,
                        java.lang.String level2Object,
                        java.lang.String level3Object,
                        java.lang.String level4Object,
                        java.lang.String notificationId,
                        java.lang.String objectType,
                        java.lang.String objectTypeQualifier,
                        java.lang.String emsTime,
                        java.lang.String neTime,
                        java.lang.String edgePointRelated,
                        java.lang.String[] attributList)

setActive

public void setActive(boolean bool)

setClient

public void setClient(org.tmforum.mtnm.nmsSession.NmsSession_I client)

Sets the client.

Parameters:

client - the new client

setModel

public void setModel(Model model)

setupNotificationRoute

public void setupNotificationRoute()

unacknowledgeAlarms

public java.util.ArrayList unacknowledgeAlarms(java.util.ArrayList alarms,
                        java.util.ArrayList addInfo)
updateEMS

public void updateEMS(java.lang.String emsName,
                      java.lang.String newEmsName,
                      java.lang.String userLabel,
                      java.lang.String nativeEMSName,
                      java.lang.String owner,
                      java.lang.String emsVersion,
                      java.lang.String type,
                      java.lang.String[] additionalInfo)

com.ericsson.nbiImp

Class TestStarter

java.lang.Object
   +--com.ericsson.nbiImp.TestStarter

public class TestStarter
extends java.lang.Object

The Class TestStarter.

Constructors

TestStarter

public TestStarter()

Methods

main

public static void main(java.lang.String[] args)

   The main method.
   Parameters:
      args - the arguments
Package com.ericsson.pojoImp

Class Summary

**ASAPIterator_IPOAImp**
The Class ASAPIterator_IPOAImp.

**BackupIdIterator_IPOAImp**
The Class BackupIdIterator_IPOAImp.

**CCIterator_IPOAImp**
The Class CCIterator_IPOAImp.

**CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp**
The Class CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp.

**CallAndTopLevelConnectionsIterator_IPOAImp**
The Class CallAndTopLevelConnectionsIterator_IPOAImp.

**Common_IPOAImp**
The Class Common_IPOAImp.

**CurrentMaintenanceOperationIterator_IPOAImp**
The Class CurrentMaintenanceOperationIterator_IPOAImp.

**EMSMgr_IPOAImp**
The Class EMSMgr_IPOAImp.

**EProtectionGroupIterator_IPOAImp**
The Class EProtectionGroupIterator_IPOAImp.

**EmsSessionFactory_IPOAImp**
The Class EmsSessionFactory_IPOAImp.

**EmsSession_IPOAImp**
The Class EmsSession_IPOAImp.

**EquipmentInventoryMgr_IPOAImp**
The Class EquipmentInventoryMgr_IPOAImp.

**EquipmentOrHolderIterator_IPOAImp**
The Class EquipmentOrHolderIterator_IPOAImp.

**EventIterator_IPOAImp**
The Class EventIterator_IPOAImp.

**FDFrIterator_IPOAImp**
The Class FDFrIterator_IPOAImp.

**FDIterator_IPOAImp**
The Class FDIterator_IPOAImp.
The Class FDIterator_IPOAlmp.

FlowDomainMgr_IPOAlmp
The Class FlowDomainMgr_IPOAlmp.

GTPiterator_IPOAlmp
The Class GTPiterator_IPOAlmp.

GuiCutThroughMgr_IPOAlmp
The Class GuiCutThroughMgr_IPOAlmp.

MDFrIterator_IPOAlmp
The Class MDFrIterator_IPOAlmp.

MDFIterator_IPOAlmp
The Class MDFIterator_IPOAlmp.

MLSNPPIterator_IPOAlmp
The Class MLSNPPIterator_IPOAlmp.

MLSNPPLinkIterator_IPOAlmp
The Class MLSNPPLinkIterator_IPOAlmp.

MLSNPPLinkMgr_IPOAlmp
The Class MLSNPPLinkMgr_IPOAlmp.

MLSNPPMgr_IPOAlmp
The Class MLSNPPPMgr_IPOAlmp.

MaintenanceMgr_IPOAlmp
The Class MaintenanceMgr_IPOAlmp.

ManagedElementIterator_IPOAlmp
The Class ManagedElementIterator_IPOAlmp.

ManagedElementMgr_IPOAlmp
The Class ManagedElementMgr_IPOAlmp.

MultiLayerSubnetworkMgr_IPOAlmp
The Class MultiLayerSubnetworkMgr_IPOAlmp.

NamingAttributesIterator_IPOAlmp
The Class NamingAttributesIterator_IPOAlmp.

NmsSession_IPOAlmp
The Class NmsSession_IPOAlmp.

PMDataIterator_IPOAlmp
The Class PMDataIterator_IPOAlmp.

PMPIterator_IPOAlmp
The Class PMPIterator_IPOAlmp.

PerformanceManagementMgr_IPOAlmp
The Class PerformanceManagementMgr_IPOAlmp.

ProtectionGroupIterator_IPOAlmp
The Class ProtectionGroupIterator_IPOAlmp.

ProtectionMgr_IPOAlmp
The Class ProtectionMgr_IPOAlmp.

SNCIterator_IPOAlmp
The Class SNCIterator_IPOAlmp.

Session_IPOAlmp
The Class Session_IPOAlmp.

SoftwareAndDataMgr_IPOAlmp
The Class SoftwareAndDataMgr_IPOAlmp.

SubnetworkIterator_IPOAlmp
The Class SubnetworkIterator_IPOAlmp.

TCAParameterProfileIterator_IPOAlmp
The Class TCAParameterProfileIterator_IPOAlmp.

TCProfileIterator_IPOAlmp
The Class TCProfileIterator_IPOAlmp.

TCProfileMgr_IPOAlmp
The Class TCProfileMgr_IPOAlmp.

TerminationPointIterator_IPOAlmp
The Class TerminationPointIterator_IPOAlmp.

TopologicalLinkIterator_IPOAlmp
The Class TopologicalLinkIterator_IPOAlmp.

TrafficDescriptorIterator_IPOAlmp
The Class TrafficDescriptorIterator_IPOAlmp.

TrafficDescriptorMgr_IPOAlmp
The Class TrafficDescriptorMgr_IPOAlmp.

TransmissionDescriptorIterator_IPOAlmp
The Class TransmissionDescriptorIterator_IPOAlmp.

TransmissionDescriptorMgr_IPOAlmp
The Class TransmissionDescriptorMgr_IPOAlmp.

Version_IPOAlmp
This class implements part of the TMF814 SS 3.5.

**com.ericsson.poaImp**

**Class ASAPIterator_IPOAImp**

```java
java.lang.Object
    +--org.omg.PortableServer.Servant
        +--org.tmforum.mtnm.aSAP.ASAPIterator_IPOA
            +--com.ericsson.poaImp.ASAPIterator_IPOAImp
```

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler, org.tmforum.mtnm.aSAP.ASAPIterator_IOperations

< Constructors > < Methods >

**Constructors**

ASAPIterator_IPOAImp

```java
public ASAPIterator_IPOAImp()```

**Methods**

**destroy**

```java
public void destroy()```

**getLength**

```java
public int getLength()```

**next_n**

```java
public boolean next_n(int howMany, 
                      org.tmforum.mtnm.aSAP.ASAPList_THolder aSAPList)```
com.ericsson.poalmp

Class BackupIdIterator_IPOAImp

java.lang.Object
    |--org.omg.PortableServer.Servant
        |--org.tmforum.mtnm.softwareAndDataManager.BackupIdIterator_IPOA
            |--com.ericsson.poalmp.BackupIdIterator_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
    org.tmforum.mtnm.softwareAndDataManager.BackupIdIterator_IOperations

< Constructors > < Methods >

class BackupIdIterator_IPOAImp extends org.tmforum.mtnm.softwareAndDataManager.BackupIdIterator_IPOA

The Class BackupIdIterator_IPOAImp.

Constructors

BackupIdIterator_IPOAImp

public BackupIdIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany,
                        org.tmforum.mtnm.softwareAndDataManager.BackupIdList_THolder backupList)
Class CCIterator_IPOAImp

java.lang.Object
   |--org.omg.PortableServer.Servant
      |--org.tmforum.mtnm.subnetworkConnection.CCIterator_IPOA
         |--com.ericsson.poaImp.CCIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.subnetworkConnection.CCIterator_IOperations

< Constructors > < Methods >

public class CCIterator_IPOAImp
extends org.tmforum.mtnm.subnetworkConnection.CCIterator_IPOA

The Class CCIterator_IPOAImp.

Constructors

CCIterator_IPOAImp

public CCIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany,
org.tmforum.mtnm.subnetworkConnection.CrossConnectList_THolder ccList)
com.ericsson.poaImp

Class
CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp

java.lang.Object

   |--org.omg.PortableServer.Servant
   |   |--org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IPOA
   |   |   |--com.ericsson.poaImp.CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IOperations

< Constructors > < Methods >

public class CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp extends org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IPOA

The Class CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp.

Constructors

CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp

public CallAndTopLevelConnectionsAndSNCsIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()
public boolean next_n(int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsList_THolder
callAndTopLevelConnectionsAndSNCsList)
public boolean next_n(int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsList_THolder
callAndTopLevelConnectionsList)

com.ericsson.poaImp

Class Common_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.common.Common_IPOA
         +--com.ericsson.poaImp.Common_IPOAImp

All Implemented Interfaces:

< Constructors > < Methods >

public class Common_IPOAImp
extends org.tmforum.mtnm.common.Common_IPOA

The Class Common_IPOAImp.

Constructors

Common_IPOAImp

public Common_IPOAImp()

Methods

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
setAdditionalInfo
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setNativeEMSName
public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)

setOwner
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String owner)

setUserLabel
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String userLabel, boolean enforceUniqueness)

com.ericsson.poaImp

Class CurrentMaintenanceOperationIterator_IPOAImp
java.lang.Object
  |---org.omg.PortableServer.Servant
    |---org.tmforum.mtnm.maintenanceOps.CurrentMaintenanceOperationIterator_IPOA
      |---com.ericsson.poaImp.CurrentMaintenanceOperationIterator_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.maintenanceOps.CurrentMaintenanceOperationIterator_IOperations

< Constructors > < Methods >
public class CurrentMaintenanceOperationIterator_IPOAImp
extends org.tmforum.mtnm.maintenanceOps.CurrentMaintenanceOperationIterator_IPOA

The Class CurrentMaintenanceOperationIterator_IPOAImp.

### Constructors

**CurrentMaintenanceOperationIterator_IPOAImp**

public **CurrentMaintenanceOperationIterator_IPOAImp**()

### Methods

**destroy**

public void **destroy**()

**getLength**

public int **getLength**()

**next_n**

public boolean **next_n**(int howMany,
org.tmforum.mtnm.maintenanceOps.CurrentMaintenanceOperationList_THolder cMOList)

---

com.ericsson.poaImp

**Class EMSMgr_IPOAImp**

java.lang.Object

    +--org.omg.PortableServer.Servant

        +--org.tmforum.mtnm.emsMgr.EMSMgr_IPOA

            +--com.ericsson.poaImp.EMSMgr_IPOAImp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler, org.tmforum.mtnm.emsMgr.EMSMgr_IOperations
public class **EMSMgr_IPOAImp**
extends org.tmforum.mtnm.emsMgr.EMSMgr_IPOA

The Class EMSMgr_IPOAImp.

**Author:**
emikrie

### Constructors

**EMSMgr_IPOAImp**

public **EMSMgr_IPOAImp** ()

### Methods

#### acknowledgeAlarms

public void **acknowledgeAlarms** (org.tmforum.mtnm.notifications.AlarmOrTCAIdentifier_T[] acknowledgeIDList,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalInfo,
org.tmforum.mtnm.notifications.AlarmAndTCAIDList_THolder failedAcknowledgeIDList)

#### assignASAP

public void **assignASAP** (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] aSAPName,
resourceName, short layerRate,
org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

#### createASAP

public void **createASAP** (org.tmforum.mtnm.aSAP.ASAPCreateModifyData_T newASAPCreateData,
or tmforum.mtnm.aSAP.ASAP_THolder newASAP,
or tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
createTopologicalLink

public void createTopologicalLink
    (org.tmforum.mtnm.topologicalLink.TLCreateData_T newTLCreateData,
     org.tmforum.mtnm.topologicalLink.TopologicalLink_THolder newTopologicalLink)

deassignASAP

public void deassignASAP
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] resourceName,
     short layerRate,
     org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

deleteASAP

public void deleteASAP
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] aSAPName,
     org.tmforum.mtnm.globaldefs.NVSLList_THolder additionalInfo)

deleteTopologicalLink

public void deleteTopologicalLink
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] topoLinkName)

getASAP

public void getASAP
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] aSAPName,
     org.tmforum.mtnm.aSAP.ASAP_THolder anASAP)

getASAPAssociatedResourceNames

public void getASAPAssociatedResourceNames
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] aSAPName,
     int howMany,
     org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
     org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
getASAPbyResource

public void getASAPbyResource(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] resourceName,
short[] layerRateList,
int howMany,
org.tmforum.mtnm.aSAP.ASAPList_THolder aSAPList,
org.tmforum.mtnm.aSAP.ASAPIterator_IHolder asapIt)

getAllASAPNames

public void getAllASAPNames(int howMany,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllASAPs

public void getAllASAPs(int howMany,
org.tmforum.mtnm.aSAP.ASAPList_THolder aSAPList,
org.tmforum.mtnm.aSAP.ASAPIterator_IHolder asapIt)

getAllEMSAndMEActiveAlarms

public void getAllEMSAndMEActiveAlarms(java.lang.String[] excludeProbCauseList,
org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
int howMany,
org.tmforum.mtnm.notifications.EventList_THolder eventList,
org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)

getAllEMSAndMEUnacknowledgedActiveAlarms

public void getAllEMSAndMEUnacknowledgedActiveAlarms(java.lang.String[] excludeProbCauseList,
org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
int howMany,
org.tmforum.mtnm.notifications.EventList_THolder eventList,
org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)
**getAllEMSSystemActiveAlarms**

```java
public void getAllEMSSystemActiveAlarms(org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
                                        int howMany,
                                        org.tmforum.mtnm.notifications.EventList_THolder eventList,
                                        org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)
```

**getAllEMSSystemUnacknowledgedActiveAlarms**

```java
public void getAllEMSSystemUnacknowledgedActiveAlarms(org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
                                                     int howMany,
                                                     org.tmforum.mtnm.notifications.EventList_THolder eventList,
                                                     org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)
```

**getAllMLRAs**

```java
public void getAllMLRAs(int howMany,
                        org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkList_THolder mLRAList,
                        org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IHolder sIt)
```

**getAllMLSNPPLinks**

```java
public void getAllMLSNPPLinks(boolean sNPListRequested,
                               int howMany,
                               org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
                               org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)
```

**getAllMLSNPPLinksWithMLSNs**

```java
public void getAllMLSNPPLinksWithMLSNs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] mLRANames,
                                        boolean sNPListRequested,
                                        int howMany,
                                        org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
                                        org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)
```
getAllMLSNPPLinksWithTNAs

```java
public void getAllMLSNPPLinksWithTNAs(java.lang.String[] tNAList,
                                       boolean sNPListRequested,
                                       int howMany,
                                       org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
                                       org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)
```

g.getAllMLSNPPLinksWithTP

```java
public void getAllMLSNPPLinksWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tPName,
                                      boolean sNPListRequested,
                                      int howMany,
                                      org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
                                      org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)
```

g.getAllMLSNPPs

```java
public void getAllMLSNPPs(boolean sNPListRequested,
                           int howMany,
                           org.tmforum.mtnm.mLSNPP.MLSNPPList_THolder mLSNPPList,
                           org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IHolder mLSNPPIt)
```

g.getAllMLSNPPsWithTNA

```java
public void getAllMLSNPPsWithTNA(java.lang.String[] tNAList,
                                 boolean sNPListRequested,
                                 int howMany,
                                 org.tmforum.mtnm.mLSNPP.MLSNPPList_THolder mLSNPPList,
                                 org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IHolder mLSNPPIt)
```

g.getAllMLSNPPsWithTP

```java
public void getAllMLSNPPsWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tPName,
                                 boolean sNPListRequested,
                                 int howMany,
                                 org.tmforum.mtnm.mLSNPP.MLSNPPList_THolder mLSNPPList,
                                 org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IHolder mLSNPPIt)
```
getAllTopLevelSubnetworkNames
public void getAllTopLevelSubnetworkNames(int howMany,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllTopLevelSubnetworks
public void getAllTopLevelSubnetworks(int howMany,
org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkList_THolder sList,
org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IHolder sIt)

getAllTopLevelTopologicalLinkNames
public void getAllTopLevelTopologicalLinkNames(int howMany,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllTopLevelTopologicalLinks
public void getAllTopLevelTopologicalLinks(int howMany,
org.tmforum.mtnm.topologicalLink.TopologicalLinkList_THolder topoList,
org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IHolder topoIt)

capabilities
public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getEMS
public void getEMS(org.tmforum.mtnm.emsMgr.EMS_THolder emsInfo)
getTopLevelTopologicalLink

public void getTopLevelTopologicalLink(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] topoLinkName,
                                      org.tmforum.mtnm.topologicalLink.TopologicalLink_THolder topoLink)

modifyASAP

public void modifyASAP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] aSAPName,
aSAPModifyData, org.tmforum.mtnm.aSAP.ASAPCreateModifyData_T
aSAPModifyData,
org.tmforum.mtnm.aSAP.ASAP_THolder newASAP,
org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
additionalInfo)

setNBI

public void setNBI(NBIImp nbi)

Sets the NBI imp. It only exists max one NBI per protocol at any given time, this nbi is given to this
manager under the given session by this function.

Parameters:

nbi - the new NBI imp. Used for debugging and logging, and to communicate with the
actual simulator.

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
java.lang.String nativeEMSName)
**setOwner**

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)
```

**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)
```

**unacknowledgeAlarms**

```java
public void unacknowledgeAlarms(org.tmforum.mtnm.notifications.AlarmOrTCAIdentifier_T[] unacknowledgeIDList,
                                 org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalInfo,
                                 org.tmforum.mtnm.notifications.AlarmAndTCAIDList_THolder failedunAcknowledgeIDList)
```

---

**com.ericsson.poaImp**

**Class EProtectionGroupIterator_IPOAImp**

```
java.lang.Object
   |--org.omg.PortableServer.Servant
      |--org.tmforum.mtnm.protection.EProtectionGroupIterator_IPOA
         |--com.ericsson.poaImp.EProtectionGroupIterator_IPOAImp
```

*All Implemented Interfaces:*
- org.omg.CORBA.portable.InvokeHandler
- org.tmforum.mtnm.protection.EProtectionGroupIterator_IOperations

**Constructors**

```java
public class EProtectionGroupIterator_IPOAImp
extends org.tmforum.mtnm.protection.EProtectionGroupIterator_IPOA
```

The Class EProtectionGroupIterator_IPOAImp.
EProtectionGroupIterator_IPOAImp

public EProtectionGroupIterator_IPOAImp()

### Methods

**destroy**

class: public void destroy()

**getLength**

class: public int getLength()

**next_n**

class: public boolean next_n(int howMany,
org.tmforum.mtnm.protection.EProtectionGroupList_THolder ePGPList)

---

**com.ericsson.poalmp**

**Class EmsSessionFactory_IPOAImp**

java.lang.Object

  +--org.omg.PortableServer.Servant

    +--org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_IPOA

      +--com.ericsson.poalmp.EmsSessionFactory_IPOAImp

All Implemented Interfaces:
- org.omg.CORBA.portable.InvokeHandler,
- org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_IOperations

< Constructors > < Methods >

public class EmsSessionFactory_IPOAImp

extends org.tmforum.mtnm.emsSessionFactory.EmsSessionFactory_IPOA

The Class EmsSessionFactory_IPOAImp. Entry point for TMF814.

Author:

   emikrie
Constructors

EmsSessionFactory_IPOAImp

public EmsSessionFactory_IPOAImp()

Methods

getEmsSession

public void getEmsSession(java.lang.String user,
java.lang.String password,
org.tmforum.mtnm.nmsSession.NmsSession_I client,
org.tmforum.mtnm.emsSession.EmsSession_IHolder
emsSessionInterface)

getVersion

public java.lang.String getVersion()

setNBIImp

public void setNBIImp(NBIImp nbi)

Sets the NBI imp. The NBI is only needed for debugging purposes and logging in this class.
Parameters:
nbi - the new Instance of the NBI

com.ericsson.poaImp

Class EmsSession_IPOAImp

java.lang.Object
|--org.omg.PortableServer.Servant
|   --org.tmforum.mtnm.emsSession.EmsSession_IPOA
|       --com.ericsson.poaImp.EmsSession_IPOAImp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.emsSession.EmsSession_IOperations
public class EmsSession_IPOAImp
extends org.tmforum.mtnm.emsSession.EmsSession_IPOA

The Class EmsSession_IPOAImp.

Author:
  emikrie

Constructors

EmsSession_IPOAImp

public EmsSession_IPOAImp()

Methods

associateSession

public void associateSession(org.tmforum.mtnm.nmsSession.NmsSession_I client)
   
   Associate session.
   
   Parameters:
   client - the client

associatedSession

public org.tmforum.mtnm.session.Session_I associatedSession()

endSession

public void endSession()

getEventChannel

public void getEventChannel(org.omg.CosNotifyChannelAdmin.EventChannelHolder eventChannel)
getManager

public void getManager(java.lang.String managerName,
org.tmforum.mtnm.common.Common_IHolder
managerInterface)

getSupportedManagers

public void getSupportedManagers(org.tmforum.mtnm.emsSession.EmsSession_IPackage.managerNames_THolder
supportedManagerList)

ping

public void ping()

setNBIImp

public void setNBIImp(NBImp nbi)

Sets the NBI imp. It only exists max one NBI per protocol at any given time, this nbi is given to this
session by this function.

Parameters:

nbi - the new NBI imp. Used for debugging and logging, and is sent to the manager that
needs it to perform specific tasks on the database.

com.ericsson.poaImp

Class EquipmentInventoryMgr_IPOAImp

java.lang.Object

|--org.omg.PortableServer.Servant
  |--org.tmforum.mtnm.equipment.EquipmentInventoryMgr_IPOA
    |--com.ericsson.poaImp.EquipmentInventoryMgr_IPOAImp

All Implemented Interfaces:

org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.equipment.EquipmentInventoryMgr_IOperations

< Constructors > < Methods >

public class EquipmentInventoryMgr_IPOAImp
extends org.tmforum.mtnm.equipment.EquipmentInventoryMgr_IPOA
The Class EquipmentInventoryMgr_IPOAImp.

## Constructors

**EquipmentInventoryMgr_IPOAImp**

public `EquipmentInventoryMgr_IPOAImp()`

## Methods

### getAllEquipment

public void `getAllEquipment` (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meOrHolderName, int howMany, org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder eqList, org.tmforum.mtnm.equipment.EquipmentOrHolderIterator_IHolder eqIt)

### getAllEquipmentNames

public void `getAllEquipmentNames` (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meOrHolderName, int howMany, org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList, org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

### getAllSupportedPTPNames

public void `getAllSupportedPTPNames` (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] equipmentName, int howMany, org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList, org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
getAllSupportedPTPs

public void
getAllSupportedPTPs (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
    equipmentName,
    int howMany,
    org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
    org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)

getAllSupportingEquipment

public void
getAllSupportingEquipment (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
    ptpOrMfdName,
    org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder eqList)

ggetAllSupportingEquipmentNames

public void
ggetAllSupportingEquipmentNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
    ptpOrMfdName,
    org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList)

ggetCapabilities

public void getCapabilities (org.tmforum.mtnm.common.CapabilityList_THolder
capabilities)

ggetContainedEquipment

public void
ggetContainedEquipment (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
equipmentHolderName,
    org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder
equipmentOrHolderList)

ggetEquipment

public void getEquipment (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
equipmentOrHolderName,
    org.tmforum.mtnm.equipment.EquipmentOrHolder_THolder
equip)
getSupportedEquipment

```java
public void getSupportedEquipment (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] equipmentName,
org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder eqList)
```

getSupportedEquipmentNames

```java
public void getSupportedEquipmentNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
equipmentName,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList)
```

getSupportingEquipment

```java
public void getSupportingEquipment (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
equipmentName,
org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder eqList)
```

getSupportingEquipmentNames

```java
public void getSupportingEquipmentNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
equipmentName,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList)
```

provisionEquipment

```java
public void provisionEquipment (org.tmforum.mtnm.equipment.EQTCreateData_T
equipmentCreateData,
org.tmforum.mtnm.equipment.Equipment_THolder createdEquipment)
```
setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setAlarmReportingOff

public void setAlarmReportingOff(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] equipmentOrHolderName)

setAlarmReportingOn

public void setAlarmReportingOn(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] equipmentOrHolderName)

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String owner)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String userLabel, boolean enforceUniqueness)
public void unprovisionEquipment(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] equipmentName)

com.ericsson.poaImp

Class EquipmentOrHolderIterator_IPOAImp

java.lang.Object
  |-- org.omg.PortableServer.Servant
    |-- org.tmforum.mtnm.equipment.EquipmentOrHolderIterator_IPOA
      |-- com.ericsson.poaImp.EquipmentOrHolderIterator_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.equipment.EquipmentOrHolderIterator_IOperations

< Constructors > < Methods >

class EquipmentOrHolderIterator_IPOAImp
    extends org.tmforum.mtnm.equipment.EquipmentOrHolderIterator_IPOA

The Class EquipmentOrHolderIterator_IPOAImp.

Constructors

public EquipmentOrHolderIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()
next_n

public boolean next_n(int howMany, org.tmforum.mtnm.equipment.EquipmentOrHolderList_THolder equipmentOrHolderList)

com.ericsson.poalmp

Class EventIterator_IPOAlmp

java.lang.Object
   +--org.omg.PortableServer.Servant
        +--org.tmforum.mtnm.notifications.EventIterator_IPOA
            +--com.ericsson.poalmp.EventIterator_IPOAlmp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler,
or.tmforum.mtnm.notifications.EventIterator_IOperations

< Constructors > < Methods >

public class EventIterator_IPOAlmp
extends org.tmforum.mtnm.notifications.EventIterator_IPOA

The Class EventIterator_IPOAlmp.

Constructors

EventIterator_IPOAlmp

public EventIterator_IPOAlmp(java.util.ArrayList list, NBImp nbi, int size)

   Instantiates a new event iterator_ ipoa imp.

   Parameters:
      list - the list
      nbi - the nbi
      size - the size

Methods
destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany, org.tmforum.mtnm.notifications.EventList_THolder eventList)

com.ericsson.poaImp

Class FDFrIterator_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.flowDomainFragment.FDFrIterator_IPOA
         +--com.ericsson.poaImp.FDFrIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.flowDomainFragment.FDFrIterator_IOperations

< Constructors > < Methods >

public class FDFrIterator_IPOAImp
extends org.tmforum.mtnm.flowDomainFragment.FDFrIterator_IPOA

The Class FDFrIterator_IPOAImp.

Constructors

FDFrIterator_IPOAImp

public FDFrIterator_IPOAImp()
destroy
public void destroy()

getLength
public int getLength()

next_n
public boolean next_n(int howMany, org.tmforum.mtnm.flowDomainFragment.FDFrList_THolder fdfrList)

com.ericsson.poaImp

Class FDIterator_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.flowDomain.FDIterator_IPOA
         +--com.ericsson.poaImp.FDIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler, org.tmforum.mtnm.flowDomain.FDIterator_IOperations

< Constructors > < Methods >

public class FDIterator_IPOAImp
extends org.tmforum.mtnm.flowDomain.FDIterator_IPOA

The Class FDIterator_IPOAImp.

Constructors

FDIterator_IPOAImp
public FDIterator_IPOAImp()
**destroy**

public void destroy()

**getLength**

public int getLength()

**next_n**

public boolean next_n(int howMany,
                       org.tmforum.mtnm.flowDomain.FDList_THolder fdList)

---

**com.ericsson.poaImp**

**Class FlowDomainMgr_IPOAImp**

java.lang.Object
   +--org.omg.PortableServer.Servant
       +--org.tmforum.mtnm.flowDomain.FlowDomainMgr_IPOA
           +--com.ericsson.poaImp.FlowDomainMgr_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.flowDomain.FlowDomainMgr_IOperations

**Constructors**

public class FlowDomainMgr_IPOAImp
extends org.tmforum.mtnm.flowDomain.FlowDomainMgr_IPOA

The Class FlowDomainMgr_IPOAImp.

**Methods**

**FlowDomainMgr_IPOAImp**

public FlowDomainMgr_IPOAImp()
**assignCPTPsToMFD**

```java
public void assignCPTPsToMFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] tpNames,
org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
org.omg.CORBA.StringHolder errorReason)
```

**associateCPTPsWithFlowDomain**

```java
public void associateCPTPsWithFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] cptpNames,
org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
org.omg.CORBA.StringHolder errorReason)
```

**associateMFDsWithFlowDomain**

```java
public void associateMFDsWithFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] mfdNames,
org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
org.omg.CORBA.StringHolder errorReason)
```

**createAndActivateFDFr**

```java
public void createAndActivateFDFr(org.tmforum.mtnm.flowDomainFragment.FDFrCreateData_T
createData,
org.tmforum.mtnm.flowDomain.ConnectivityRequirement_T connectivityRequirement,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] aEnd,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] zEnd,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder internalTPs,
org.tmforum.mtnm.flowDomainFragment.MatrixFlowDomainFragmentList_THolder mfdfrs,
org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
org.tmforum.mtnm.flowDomainFragment.FlowDomainFragment_THolder theFDFr,
notConnectableCPTPList,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder parameterProblemsTPList,
org.omg.CORBA.StringHolder errorReason)
```
createFTP

public void createFTP(org.tmforum.mtnm.flowDomain.FTPCreateData_T createData,
                       org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                       org.tmforum.mtnm.terminationPoint.TerminationPoint_THolder theFTP,
                       org.omg.CORBA.StringHolder errorReason)

createFlowDomain

public void createFlowDomain(org.tmforum.mtnm.flowDomain.FDCreateData_T createData,
                             org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder assignedCPTPs,
                             org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                             org.tmforum.mtnm.flowDomain.FlowDomain_THolder theFD,
                             org.omg.CORBA.StringHolder errorReason)

createMFD

public void createMFD(org.tmforum.mtnm.flowDomain.MFDCreateData_T createData,
                      org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                      org.tmforum.mtnm.flowDomain.MatrixFlowDomain_THolder theMFD,
                      org.omg.CORBA.StringHolder errorReason)

deAssociateCPTPsFromFlowDomain

public void deAssociateCPTPsFromFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                                            org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] tpNames,
                                            org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                                            org.omg.CORBA.StringHolder errorReason)

deAssociateMFDsFromFlowDomain

public void deAssociateMFDsFromFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                                          org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] mfdNames,
                                          org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                                          org.omg.CORBA.StringHolder errorReason)
**deactivateAndDeleteFDFr**

```java
public void deactivateAndDeleteFDFr(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdfrName,
                                   org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                                   org.tmforum.mtnm.flowDomainFragment.FlowDomainFragment_THolder theFDFr,
                                   org.omg.CORBA.StringHolder errorReason)
```

**deleteFTP**

```java
public void deleteFTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] ftpName,
                      org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                      org.omg.CORBA.StringHolder errorReason)
```

**deleteFlowDomain**

```java
public void deleteFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                             org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                             org.omg.CORBA.StringHolder errorReason)
```

**deleteMFD**

```java
public void deleteMFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
                      org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                      org.omg.CORBA.StringHolder errorReason)
```

**getAllAssignableCPTPs**

```java
public void getAllAssignableCPTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
                                   int howMany,
                                   org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder cptpList,
                                   org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder cctpIt)
```
getAllAssignedCPTPs

public void getAllAssignedCPTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
                               int howMany,
                               org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder cptpList,
                               org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder cptpIt)

getAllAssociatedMFDs

public void getAllAssociatedMFDs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tmdOrFdName,
                                  int howMany,
                                  org.tmforum.mtnm.flowDomain.MFDList_THolder mfds,
                                  org.tmforum.mtnm.flowDomain.MFDIterator_IHolder mfdIt)

ggetAllCPTPs

public void getAllCPTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                         org.tmforum.mtnm.flowDomain.CPTP_Role_T cptpRole,
                         int howMany,
                         org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder cptpList,
                         org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder cptpIt)

ggetAllFDFrs

public void getAllFDFrs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                         int howMany,
                         short[] connectivityRateList,
                         org.tmforum.mtnm.flowDomainFragment.FDFrList_THolder fdfrList,
                         org.tmforum.mtnm.flowDomainFragment.FDFrIterator_IHolder fdfrIt)

ggetAllFlowDomains

public void getAllFlowDomains(int howMany,
                              org.tmforum.mtnm.flowDomain.FDList_THolder flowDomains,
                              org.tmforum.mtnm.flowDomain.FDIterator_IHolder fdIt)
getAllSupportedMFDs

public void getAllSupportedMFDs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] holderName,
int howMany,
org.tmforum.mtnm.flowDomain.MFDList_THolder mfds,
org.tmforum.mtnm.flowDomain.MFDIterator_IHolder mfdIt)

getAllTopologicalLinksOfFD

public void getAllTopologicalLinksOfFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] flowDomainName,
int howMany,
org.tmforum.mtnm.topologicalLink.TopologicalLinkList_THolder topoList,
org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IHolder topoIt)

getAssigningMFD

public void getAssigningMFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] cptpName,
org.tmforum.mtnm.flowDomain.MatrixFlowDomain_THolder mfd)

getAssociatingFD

public void getAssociatingFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
org.tmforum.mtnm.flowDomain.FlowDomain_THolder flowDomain)

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
getFDFr

public void getFDFr
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdfrName,
org.tmforum.mtnm.flowDomainFragment.FlowDomainFragment_THolder fdfr)

getFDFrRoute

public void getFDFrRoute
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdfrName,
org.tmforum.mtnm.flowDomainFragment.FDFrRoute_THolder route)

getFDFrsByUserLabel

public void getFDFrsByUserLabel
(java.lang.String userLabel,
org.tmforum.mtnm.flowDomainFragment.FDFrList_THolder fdfrs)

getFDFrsWithTP

public void getFDFrsWithTP
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] cptpName,
int howMany,
org.tmforum.mtnm.flowDomainFragment.FDFrList_THolder fdfrList,
org.tmforum.mtnm.flowDomainFragment.FDFrIterator_IHolder fdfrIt)

getFlowDomain

public void getFlowDomain
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
org.tmforum.mtnm.flowDomain.FlowDomain_THolder flowDomain)

getFlowDomainsByUserLabel

public void getFlowDomainsByUserLabel
(java.lang.String userLabel,
org.tmforum.mtnm.flowDomain.FDList_THolder flowDomains)
getMFD
public void getMFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
                   org.tmforum.mtnm.flowDomain.MatrixFlowDomain_THolder mfd)

getTransmissionParams
public void getTransmissionParams(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] name,
                                   java.lang.String[] filter,
                                   org.tmforum.mtnm.transmissionParameters.LayeredParameterList_THolder transmissionParams)

modifyFDFr
public void modifyFDFr(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdfrName,
                        org.tmforum.mtnm.flowDomainFragment.FDFrModifyData_T fdfrModifyData,
                        org.tmforum.mtnm.flowDomain.ConnectivityRequirement_T connectivityRequirement,
                        org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                        org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder failedTPList,
                        org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder parameterProblemsTPList,
                        org.tmforum.mtnm.flowDomainFragment.FlowDomainFragment_THolder newFDFr,
                        org.omg.CORBA.StringHolder errorReason)

modifyFlowDomain
public void modifyFlowDomain(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fdName,
                              org.tmforum.mtnm.flowDomain.FDModifyData_T fdModifyData,
                              org.tmforum.mtnm.flowDomain.FlowDomain_THolder modifiedFD,
                              org.omg.CORBA.StringHolder failedAttributes,
                              org.omg.CORBA.StringHolder errorReason)
modifyMFD

public void modifyMFD(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName,
                      org.tmforum.mtnm.flowDomain.MFDModifyData_T mfdModifyData,
                      org.tmforum.mtnm.flowDomain.MatrixFlowDomain_THolder newMFD,
                      org.omg.CORBA.StringHolder failedAttributes,
                      org.omg.CORBA.StringHolder errorReason)

setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                              org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                              java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)
public void unassignCPTPsFromMFD (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mfdName, org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] tpNames, org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify, org.omg.CORBA.StringHolder errorReason)

com.ericsson.poaImp

Class GTPiterator_IPOAImp

java.lang.Object
  +--org.omgPortableServer.Servant
  |    +--org.tmforum.mtnm.terminationPoint.GTPiterator_IPOA
  |        +--com.ericsson.poaImp.GTPiterator_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.terminationPoint.GTPiterator_IOperations

< Constructors > < Methods >

public class GTPiterator_IPOAImp
extends org.tmforum.mtnm.terminationPoint.GTPiterator_IPOA

The Class GTPiterator_IPOAImp.

Constructors

GTPiterator_IPOAImp

public GTPiterator_IPOAImp()  

Methods

destroy

public void destroy()
**getLength**

```
public int getLength()
```

**next_n**

```
public boolean next_n(int howMany,
                      org.tmforum.mtnm.terminationPoint.GTPlist_THolder gtpList)
```

---

`com.ericsson.poaImp`

**Class GuiCutThroughMgr_IPOAImp**

```
java.lang.Object
  +--org.omg.PortableServer.Servant
    +--org.tmforum.mtnm.guiCutThrough.GuiCutThroughMgr_IPOA
      +--com.ericsson.poaImp.GuiCutThroughMgr_IPOAImp
```

**All Implemented Interfaces:**
- org.omg.CORBA.portable.InvokeHandler
- org.tmforum.mtnm.guiCutThrough.GuiCutThroughMgr_IOperations

< Constructors > < Methods >

**public class GuiCutThroughMgr_IPOAImp**
extends org.tmforum.mtnm.guiCutThrough.GuiCutThroughMgr_IPOA

The Class GuiCutThroughMgr_IPOAImp.

---

**Constructors**

**public GuiCutThroughMgr_IPOAImp()**

---

**Methods**

**destroyGCT**

```
public void destroyGCT(java.lang.String displayAddress)
```
getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getGCTProfileInfo

public void getGCTProfileInfo(org.tmforum.mtnm.guiCutThrough.GCTProfileInfo_THolder gctProfileInfo)

launchGCT

public void launchGCT(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                        java.lang.String gctContext,
                        org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] userInput,
                        java.lang.String displayAddress,
                        org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalInputInfo,
                        org.omg.CORBA.BooleanHolder closingEnabled,
                        org.tmforum.mtnm.globaldefs.NVSList_THolder additionalOutputInfo)

setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                               org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setNBI

public void setNBI(NBIImp nbi)

    Sets the nBI.

    Parameters:

    nbi - the new nBI
### setNativeEMSName

```java
public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
                   java.lang.String nativeEMSName)
```

### setOwner

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
                   java.lang.String owner)
```

### setUserLabel

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
                   java.lang.String userLabel,
                   boolean enforceUniqueness)
```

---

**com.ericsson.poaImp**

## Class MFDFrIterator_IPOAImp

```java
java.lang.Object
|--org.omg.PortableServer.Servant
   |--org.tmforum.mtnm.flowDomainFragment.MFDFrIterator_IPOA
     |--com.ericsson.poaImp.MFDFrIterator_IPOAImp
```

*All Implemented Interfaces:*
- org.omg.CORBA.portable.InvokeHandler,
- org.tmforum.mtnm.flowDomainFragment.MFDFrIterator_IOperations

< Constructors > < Methods >

```java
public class MFDFrIterator_IPOAImp
extends org.tmforum.mtnm.flowDomainFragment.MFDFrIterator_IPOA

The Class MFDFrIterator_IPOAImp.
```
MFDFrIterator_IPOAImp

public MFDFrIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany,
org.tmforum.mtnm.flowDomainFragment.MatrixFlowDomainFragmentList_THolder
mfdfrList)

com.ericsson.poaImp

Class MFDIterator_IPOAImp

java.lang.Object

---org.omg.PortableServer.Servant

---org.tmforum.mtnm.flowDomain.MFDIterator_IPOA

---com.ericsson.poaImp.MFDIterator_IPOAImp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.flowDomain.MFDIterator_IOperations

< Constructors > < Methods >

public class MFDIterator_IPOAImp
extends org.tmforum.mtnm.flowDomain.MFDIterator_IPOA

The Class MFDIterator_IPOAImp.
MFDIterator_IPOAImp

```java
public MFDIterator_IPOAImp()
```

### Methods

**destroy**

```java
public void destroy()
```

**getLength**

```java
public int getLength()
```

**next_n**

```java
public boolean next_n(int howMany, org.tmforum.mtnm.flowDomain.MFDList_THolder mfdList)
```

---

**com.ericsson.poalmp**

**Class MLSNPPIterator_IPOAImp**

```java
java.lang.Object
decorated by:
  +--org.omg.PortableServer.Servant
        +--org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IPOA
            +--com.ericsson.poalmp.MLSNPPIterator_IPOAImp
```

*All Implemented Interfaces:*

- org.omg.CORBA.portable.InvokeHandler
- org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IOperations

< Constructors > < Methods >

```java
public class MLSNPPIterator_IPOAImp
extends org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IPOA

The Class MLSNPPIterator_IPOAImp.
```
public class MLSNPPLinkIterator_IPOAImp
extends org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IPOA

The Class MLSNPPLinkIterator_IPOAImp.
MLSNPPLinkIterator_IPOAImp

public MLSNPPLinkIterator_IPOAImp()

## Methods

**destroy**

public void destroy()

**getLength**

public int getLength()

**next_n**

public boolean next_n(int howMany,
org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder
mLSNPPLink)

## com.ericsson.poaImp

### Class MLSNPPLinkMgr_IPOAImp

java.lang.Object

|--org.omg.PortableServer.Servant

|--org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkMgr_IPOA

|--com.ericsson.poaImp.MLSNPPLinkMgr_IPOAImp

All Implemented Interfaces:
org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkMgr_IOperations

< Constructors > < Methods >

public class MLSNPPLinkMgr_IPOAImp
extends org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkMgr_IPOA

The Class MLSNPPLinkMgr_IPOAImp.
MLSNPPLinkMgr_IPOAmp

public MLSNPPLinkMgr_IPOAmp()

Methods

assignSignallingController

public void assignSignallingController(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName, java.lang.String signallingControllerIdentifier)

deassignSignallingController

public void deassignSignallingController(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName)

disableSignalling

public void disableSignalling(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName)

enableSignalling

public void enableSignalling(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName)

getAvailableCapacity

public void getAvailableCapacity(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName, short layerRate, org.tmforum.mtnm.mLSNPPLink.AvailableCapacity_THolder availableCapacity)
getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

modifySignallingProtocolParameters

public void modifySignallingProtocolParameters(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName, org.tmforum.mtnm.globaldefs.NVSLList_THolder signallingParameters)

setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSLList_THolder additionalInfo)

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String owner)

setSignallingProtocolAndParameters

public void setSignallingProtocolAndParameters(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName, java.lang.String signallingProtocol, org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] signallingParameters)
setTNANNameForMLSNPPLinkEnd

public void setTNANNameForMLSNPPLinkEnd(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName,
                                         org.tmforum.mtnm.mLSNPP.SNPTNAData_T[] sNPTNADataList,
                                         org.tmforum.mtnm.mLSNPP.SNPPTNAPair_T[] sNPPTNAPairList,
                                         java.lang.String aTNAName,
                                         java.lang.String zTNAName,
                                         java.lang.String aTNAGroupName,
                                         java.lang.String zTNAGroupName,
                                         org.tmforum.mtnm.mLSNPPLink.MultiLayerSNPPLink_THolder mLSNPPLink)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)

com.ericsson.poaImp

Class MLSNPPMgr_IPOAImp

java.lang.Object
    +--org.omg.PortableServer.Servant
        +--org.tmforum.mtnm.mLSNPP.MLSNPPMgr_IPOA
            +--com.ericsson.poaImp.MLSNPPMgr_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler, org.tmforum.mtnm.mLSNPP.MLSNPPMgr_IOperations

< Constructors > < Methods >

public class MLSNPPMgr_IPOAImp
extends org.tmforum.mtnm.mLSNPP.MLSNPPMgr_IPOA

The Class MLSNPPMgr_IPOAImp.

Constructors

MLSNPPMgr_IPOAImp

public MLSNPPMgr_IPOAImp()
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCapabilities</td>
<td>public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)</td>
</tr>
<tr>
<td>getMLSNPP</td>
<td>public void getMLSNPP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPName, org.tmforum.mtnm.mLSNPP.MultiLayerSNPP_THolder theMLSNPP)</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)</td>
</tr>
<tr>
<td>setNativeEMSName</td>
<td>public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)</td>
</tr>
<tr>
<td>setOwner</td>
<td>public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String owner)</td>
</tr>
</tbody>
</table>
**setTNANameForMLSNPP**

```java
public void setTNANameForMLSNPP (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPName,
                                 org.tmforum.mtnm.mLSNPP.SNPTNAData_T[] sNPTNADataList,
                                 org.tmforum.mtnm.mLSNPP.SNPTNAPair_T[] sNPPTNAPairList,
                                 java.lang.String tNAName,
                                 java.lang.String tNAGroupName,
                                 org.tmforum.mtnm.mLSNPP.MultiLayerSNPP_THolder mLSNPP)
```

---

**setUserLabel**

```java
public void setUserLabel (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)
```

---

**com.ericsson.poaImp**

**Class MaintenanceMgr_IPOAImp**

```java
java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.maintenanceOps.MaintenanceMgr_IPOA
         +--com.ericsson.poaImp.MaintenanceMgr_IPOAImp
```

**All Implemented Interfaces:**

- org.omg.CORBA.portable.InvokeHandler,
- org.tmforum.mtnm.maintenanceOps.MaintenanceMgr_IOperations

---

**Constructors**

**MaintenanceMgr_IPOAImp**

```java
public MaintenanceMgr_IPOAImp()
```
Methods

**getActiveMaintenanceOperations**

```java
```

**getCapabilities**

```java
getCapabilities (org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
```

**performMaintenanceOperation**

```java
```

**setAdditionalInfo**

```java
setAdditionalInfo (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
```

**setNativeEMSName**

```java
setNativeEMSName (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)
```
**setOwner**

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)
```

---

**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)
```

---

**com.ericsson.poaImp**

### Class ManagedElementIterator_IPOAImp

```java
java.lang.Object
   |---org.omg.PortableServer.Servant
      |---org.tmforum.mtnm.managedElement.ManagedElementIterator_IPOA
         |---com.ericsson.poaImp.ManagedElementIterator_IPOAImp
```

**All Implemented Interfaces:**
- org.omg.CORBA.portable.InvokeHandler
- org.tmforum.mtnm.managedElement.ManagedElementIterator_IOperations

---

**Constructors**

**ManagedElementIterator_IPOAImp**

```java
public ManagedElementIterator_IPOAImp(java.util.ArrayList list,
                                       NBIImp nbi,
                                       int size)
```

Instantiates a new managed element iterator_IPOAImp.

**Parameters:**
- list - the list
- nbi - the nbi
- size - the size
Methods

**destroy**

public void `destroy()`

**getLength**

public int `getLength()`

**next_n**

public boolean `next_n(int howMany, org.tmforum.mtnm.managedElement.ManagedElementList_THolder meList)`

com.ericsson.poalmp

**Class ManagedElementMgr_IPOAImp**

java.lang.Object
   |---org.omg.PortableServer.Servant
      |---org.tmforum.mtnm.managedElementManager.ManagedElementMgr_IPOA
         |---com.ericsson.poalmp.ManagedElementMgr_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.managedElementManager.ManagedElementMgr_IOperations

< Constructors > < Methods >

public class `ManagedElementMgr_IPOAImp`
extends org.tmforum.mtnm.managedElementManager.ManagedElementMgr_IPOA

The Class ManagedElementMgr_IPOAImp.

Constructors

**ManagedElementMgr_IPOAImp**

public `ManagedElementMgr_IPOAImp()`
createGTP

```java
public void createGTP(java.lang.String userLabel,
                      boolean forceUniqueness,
                      java.lang.String owner,
                      org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] listOfTPs,
                      org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] initialCTPname,
                      int numberOfCTPs,
                      org.tmforum.mtnm.terminationPoint.GTPEffort_T gtpEffort,
                      org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalCreationInfo,
                      org.tmforum.mtnm.terminationPoint.GTP_THolder newGTP)
```

deleteGTP

```java
public void deleteGTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] gtpName)
```

getAllActiveAlarms

```java
public void getAllActiveAlarms(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
                               java.lang.String[] excludeProbCauseList,
                               org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
                               int howMany,
                               org.tmforum.mtnm.notifications.EventList_THolder eventList,
                               org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)
```

getAllCrossConnections

```java
public void getAllCrossConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
                                    managedElementName,
                                    short[] connectionRateList,
                                    int howMany,
                                    org.tmforum.mtnm.subnetworkConnection.CrossConnectList_THolder ccList,
                                    org.tmforum.mtnm.subnetworkConnection.CCIterator_IHolder ccIt)
```
getAllFTPNames

public void getAllFTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                           short[] tpLayerRateList,
                           short[] connectionLayerRateList,
                           int howMany,
                           org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                           org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllFTPs

public void getAllFTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                        short[] tpLayerRateList,
                        short[] connectionLayerRateList,
                        int howMany,
                        org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                        org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)

ggetAllFixedCrossConnections

public void getAllFixedCrossConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                                          short[] connectionRateList,
                                          int howMany,
                                          org.tmforum.mtnm.subnetworkConnection.CrossConnectList_THolder ccList,
                                          org.tmforum.mtnm.subnetworkConnection.CCIterator_IHolder ccIt)

ggetAllGTPNames

public void getAllGTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                            short[] tpLayerRateList,
                            int howMany,
                            org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                            org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
getAllGTPs

public void getAllGTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,  
short[] tpLayerRateList,  
int howMany,  
org.tmforum.mtnm.terminationPoint.GTPlist_THolder gtpList,  
org.tmforum.mtnm.terminationPoint.GTPiterator_IHolder gtpIt)

ggetAllManagedElementNames

public void getAllManagedElementNames(int howMany,  
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,  
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

ggetAllManagedElements

public void getAllManagedElements(int howMany,  
org.tmforum.mtnm.managedElement.ManagedElementList_THolder meList,  
org.tmforum.mtnm.managedElement.ManagedElementIterator_IHolder meIt)

ggetAllPTPNames

public void getAllPTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,  
short[] tpLayerRateList,  
short[] connectionLayerRateList,  
int howMany,  
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,  
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

ggetAllPTPNamesWithoutFTPs

public void getAllPTPNamesWithoutFTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,  
short[] tpLayerRateList,  
short[] connectionLayerRateList,  
int howMany,  
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,  
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
**getAllPTPs**

```java
public void getAllPTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                       short[] tpLayerRateList,
                       short[] connectionLayerRateList,
                       int howMany,
                       org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                       org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
```

**getAllPTPsWithoutFTPs**

```java
public void getAllPTPsWithoutFTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
                                   short[] tpLayerRateList,
                                   short[] connectionLayerRateList,
                                   int howMany,
                                   org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                                   org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
```

**getAllUnacknowledgedActiveAlarms**

```java
public void getAllUnacknowledgedActiveAlarms(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
                                             java.lang.String[] excludeProbCauseList,
                                             org.tmforum.mtnm.notifications.PerceivedSeverity_T[] excludeSeverityList,
                                             int howMany,
                                             org.tmforum.mtnm.notifications.EventList_THolder eventList,
                                             org.tmforum.mtnm.notifications.EventIterator_IHolder eventIt)
```

**getCapabilities**

```java
public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
```
getContainedCurrentTPNames

public void getContainedCurrentTPNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                             short[] layerRateList,
                                             int howMany,
                                             org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                                             org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getContainedCurrentTPs

public void getContainedCurrentTPs (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                               short[] layerRateList,
                                               int howMany,
                                               org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                                               org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)

getContainedInUseTPNames

public void getContainedInUseTPNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                                short[] layerRateList,
                                                int howMany,
                                                org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                                                org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getContainedInUseTPs

public void getContainedInUseTPs (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                                short[] layerRateList,
                                                int howMany,
                                                org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                                                org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
**getContainedPotentialTPNames**

```java
public void getContainedPotentialTPNames(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
    short[] layerRateList,
    int howMany,
    org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
    org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```

**getContainedPotentialTPs**

```java
public void getContainedPotentialTPs(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
    short[] layerRateList,
    int howMany,
    org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
    org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
```

**getContainingGTP**

```java
public void getContainingGTP(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] ctpName,
    org.tmforum.mtnm.terminationPoint.GTP_THolder gtp)
```

**getContainingSubnetworkNames**

```java
public void getContainingSubnetworkNames(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
    org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder subnetNames)
```

**getContainingTPNames**

```java
public void getContainingTPNames(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
    org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder tpNameList)
```
getContainingTPs

public void getContainingTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName, org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList)

getGTP

public void getGTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] gtpName, org.tmforum.mtnm.terminationPoint.GTP_THolder gtp)

getManagedElement

public void getManagedElement(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] me)

getPotentialFixedCCs

public void getPotentialFixedCCs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] inputTP, org.tmforum.mtnm.globaldefs.NamingAttributes_THolder ContainingTP, org.tmforum.mtnm.globaldefs.NamingAttributes_THolder potentialCCList)

getTP

public void getTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName, org.tmforum.mtnm.terminationPoint.TerminationPoint_THolder tp)

modifyGTP

public void modifyGTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] gtpName, org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] tpNames, java.lang.String actionType, org.tmforum.mtnm.terminationPoint.GTP_THolder modifiedGTP)
**setAdditionalInfo**

```java
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                                org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
```

**setGtpAlarmReportingOff**

```java
public void setGtpAlarmReportingOff(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] gtpName)
```

**setGtpAlarmReportingOn**

```java
public void setGtpAlarmReportingOn(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] gtpName)
```

**setNBI**

```java
public void setNBI(NBIImp nbi)
```

Sets the nBI.

**Parameters:**

- `nbi` - the new nBI

**setNativeEMSName**

```java
public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                                java.lang.String nativeEMSName)
```

**setOwner**

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)
```
setTPData

public void setTPData(org.tmforum.mtnm.subnetworkConnection.TPData_T tpInfo,
org.tmforum.mtnm.terminationPoint.TerminationPoint_THolder modifiedTP)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
                  java.lang.String userLabel,
                  boolean enforceUniqueness)

verifyTMDAssignment

public void verifyTMDAssignment(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                org.tmforum.mtnm.terminationPoint.Directionality_T direction,
                org.omg.CORBA.StringHolder tmdAssignmentState,
                org.tmforum.mtnm.transmissionParameters.LayeredParameterList_THolder
transmissionParams,
                org.tmforum.mtnm.globaldefs.NVSList_THolder
additionalTPInfo)

com.ericsson.poalmp

Class MultiLayerSubnetworkMgr_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.multiLayerSubnetwork.MultiLayerSubnetworkMgr_IPOA
         +--com.ericsson.poaImp.MultiLayerSubnetworkMgr_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
    org.tmforum.mtnm.multiLayerSubnetwork.MultiLayerSubnetworkMgr_IOperations

< Constructors > < Methods >

public class MultiLayerSubnetworkMgr_IPOAImp
extends org.tmforum.mtnm.multiLayerSubnetwork.MultiLayerSubnetworkMgr_IPOA

The Class MultiLayerSubnetworkMgr_IPOAImp.
public MultiLayerSubnetworkMgr_IPOAImp()

### Methods

#### activateSNC

```java
public void activateSNC(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
    org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
    org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
    org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
    org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder theSNC,
    org.omg.CORBA.StringHolder errorReason)
```

#### addConnections

```java
public void addConnections(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
    org.tmforum.mtnm.subnetworkConnection.SNCCreateData_T[] connectionsToAdd,
    boolean connectionRouteReArrangementAllowed,
    org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
    org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder connectionList,
    org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder partialSNCs,
    org.omg.CORBA.StringHolder errorReason)
```

#### addRoute

```java
public void addRoute(
    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
    org.tmforum.mtnm.subnetworkConnection.RouteCreateData_T createRoute,
    org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
    org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
    org.tmforum.mtnm.subnetworkConnection.RouteDescriptor_THolder theRoute,
    org.omg.CORBA.StringHolder errorReason)
```
checkValidSNC

public void checkValidSNC(org.tmforum.mtnm.subnetworkConnection.SNCCreateData_T createData,
                        org.tmforum.mtnm.subnetworkConnection.TPData_T[] tpsToModify,
                        boolean considerResources,
                        org.omg.CORBA.BooleanHolder valid)

createAndActivateSNC

public void createAndActivateSNC(org.tmforum.mtnm.subnetworkConnection.SNCCreateData_T createData,
                                 org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                                 org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                                 org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                                 org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder theSNC,
                                 org.omg.CORBA.StringHolder errorReason)

createModifiedSNC

public void createModifiedSNC(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                               java.lang.String routeId,
                               org.tmforum.mtnm.subnetworkConnection.SNCMODifyData_T SNCModifyData,
                               org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                               org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                               org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                               org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder newSNC,
                               org.omg.CORBA.StringHolder errorReason)

createSNC

public void createSNC(org.tmforum.mtnm.subnetworkConnection.SNCCreateData_T createData,
                       org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                       org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                       org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder theSNC,
                       org.omg.CORBA.StringHolder errorReason)
createTPPool

public void createTPPool(org.tmforum.mtnm.multiLayerSubnetwork.TPPoolCreateData_T newTPPoolCreateData,
                        org.tmforum.mtnm.terminationPoint.TerminationPoint_THolder newTPPool)

deactivateAndDeleteSNC

disable public void deactivateAndDeleteSNC (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                        org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                        org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                        org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                        org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder theSNC,
                        org.omg.CORBA.StringHolder errorReason)

deactivateSNC

disable public void deactivateSNC (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                        org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                        org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                        org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                        org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder theSNC,
                        org.omg.CORBA.StringHolder errorReason)

deleteSNC

public void deleteSNC (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                        org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel)

deleteTPPool

public void deleteTPPool (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpPoolName)
establishCall

public void establishCall(org.tmforum.mtnm.callSNC.CallCreateData_T callCreateData,
    org.tmforum.mtnm.subnetworkConnection.SNCCreateData_T[] connectionCreateDataList,
    java.lang.String routeGroupsNumber,
    org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
    org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCs_THolder callAndTopLevelConnectionsAndSNCs,
    org.tmforum.mtnm.subnetworkConnection.SNCCreateDataList_THolder sNCsNotCreated,
    org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder partialSNCs,
    org.omg.CORBA.StringHolder callErrorReason)

getAllCallIdsWithSNPPOrTNAName

public void getAllCallIdsWithSNPPOrTNAName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T sNPPOrTNAName,
    org.tmforum.mtnm.callSNC.CallIdList_THolder callIdList)

getAllCallIdsWithTP

public void getAllCallIdsWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tPName,
    org.tmforum.mtnm.callSNC.CallIdList_THolder callIdList)

getAllCallsAndTopLevelConnections

public void getAllCallsAndTopLevelConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
    int howMany,
    org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsList_THolder callAndTopLevelConnectionsList,
    org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsIterator_IHolder callAndTopLevelConnectionsIt)
getAllCallsAndTopLevelConnectionsAndSNCs

public void getAllCallsAndTopLevelConnectionsAndSNCs
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsList_THolder callAndTopLevelConnectionsAndSNCsList,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IHolder callAndTopLevelConnectionsAndSNCsIt)

g.getAllCallsAndTopLevelConnectionsAndSNCsWithME

public void getAllCallsAndTopLevelConnectionsAndSNCsWithME
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsList_THolder callAndTopLevelConnectionsAndSNCsList,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IHolder callAndTopLevelConnectionsAndSNCsIt)

g.getAllCallsAndTopLevelConnectionsAndSNCsWithTP

public void getAllCallsAndTopLevelConnectionsAndSNCsWithTP
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tPName,
int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsList_THolder callAndTopLevelConnectionsAndSNCsList,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCsIterator_IHolder callAndTopLevelConnectionsAndSNCsIt)

g.getAllCallsAndTopLevelConnectionsWithME

public void getAllCallsAndTopLevelConnectionsWithME
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
int howMany,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsList_THolder callAndTopLevelConnectionsList,
org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsIterator_IHolder callAndTopLevelConnectionsIt)
```java
public void getAllEdgeMLSNPPLinks(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
        boolean sNPListRequested,
        int howMany,
        org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
        org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)

public void getAllEdgePointNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
        short[] layerRateList,
        short[] connectionLayerRateList,
        int howMany,
        org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
        org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

public void getAllEdgePoints(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
        short[] tpLayerRateList,
        short[] connectionLayerRateList,
        int howMany,
        org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
        org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)

public void getAllFixedSubnetworkConnectionNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
        short[] connectionRateList,
        int howMany,
        org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
        org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```
public void getAllFixedSubnetworkConnectionNamesWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
connectionRateList,
short[]
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

public void getAllFixedSubnetworkConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
short[] connectionRateList,
int howMany,
org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sncList,
org.tmforum.mtnm.subnetworkConnection.SNCIterator_IHolder sncIt)

public void getAllFixedSubnetworkConnectionsWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
short[] connectionRateList,
int howMany,
org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sncList,
org.tmforum.mtnm.subnetworkConnection.SNCIterator_IHolder sncIt)

public void getAllInternalMLSNPPLinks(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
boolean sNPListRequested,
int howMany,
org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)
getAllMLSNPPLinks

public void getAllMLSNPPLinks(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
  boolean sNPListRequested,
  int howMany,
  org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkList_THolder mLSNPPLinkList,
  org.tmforum.mtnm.mLSNPPLink.MLSNPPLinkIterator_IHolder mLSNPPLinkIt)

getAllMLSNPPs

public void getAllMLSNPPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
  boolean sNPListRequested,
  int howMany,
  org.tmforum.mtnm.mLSNPP.MLSNPPList_THolder mLSNPPList,
  org.tmforum.mtnm.mLSNPP.MLSNPPIterator_IHolder mLSNPPIt)

getAllManagedElementNames

public void getAllManagedElementNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
  int howMany,
  org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
  org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllManagedElements

public void getAllManagedElements(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
  int howMany,
  org.tmforum.mtnm.managedElement.ManagedElementList_THolder meList,
  org.tmforum.mtnm.managedElement.ManagedElementIterator_IHolder meIt)
getAllSubnetworkConnectionNames

public void getAllSubnetworkConnectionNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                                           short[] connectionRateList,
                                           int howMany,
                                           org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                                           org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllSubnetworkConnectionNamesWithTP

public void getAllSubnetworkConnectionNamesWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                                  short[] connectionRateList,
                                                  int howMany,
                                                  org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                                                  org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllSubnetworkConnections

public void getAllSubnetworkConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                                        short[] connectionRateList,
                                        int howMany,
                                        org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sncList,
                                        org.tmforum.mtnm.subnetworkConnection.SNCIterator_IHolder sncIt)

getAllSubnetworkConnectionsWithTP

public void getAllSubnetworkConnectionsWithTP(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                               short[] connectionRateList,
                                               int howMany,
                                               org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sncList,
                                               org.tmforum.mtnm.subnetworkConnection.SNCIterator_IHolder sncIt)
getAllSubordinateMLSNs

```java
public void getAllSubordinateMLSNs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                               int howMany,
                               org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkList_THolder subordinateMLSNsList,
                               org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IHolder subnetIt)
```

getAllSubordinateRAIdsWithConnection

```java
public void getAllSubordinateRAIdsWithConnection(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                                                 org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] connection,
                                                 java.lang.String routeType,
                                                 org.tmforum.mtnm.multiLayerSubnetwork.RoutePerRouteType_THolder routePerRouteType)
```

getAllTPPoolNames

```java
public void getAllTPPoolNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetworkName,
                               int howMany,
                               org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                               org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```

getAllTPPools

```java
public void getAllTPPools(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetworkName,
                           int howMany,
                           org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
                           org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
```

getAllTopologicalLinkNames

```java
public void getAllTopologicalLinkNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                                        int howMany,
                                        org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
                                        org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```
getAllTopologicalLinks

public void getAllTopologicalLinks(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName,
                                   int howMany,
                                   org.tmforum.mtnm.topologicalLink.TopologicalLinkList_THolder topoList,
                                   org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IHolder topoIt)

getAssociatedTP

public void getAssociatedTP (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                             org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList)

getBackupRoutes

public void getBackupRoutes(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                            java.lang.String routeId,
                            boolean includeHigherOrderCCs,
                            org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo,
                            org.tmforum.mtnm.subnetworkConnection.RouteList_THolder routeList)

call

public void getCall (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
                     org.tmforum.mtnm.callSNC.Call_THolder theCall)

getCallAndTopLevelConnections

public void getCallAndTopLevelConnections (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
                                            java.lang.String callId,
                                            org.tmforum.mtnm.callSNC.CallAndTopLevelConnections_THolder callAndTopLevelConnections)
public void getCallAndTopLevelConnectionsAndSNCs
  (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
   org.tmforum.mtnm.callSNC.CallAndTopLevelConnectionsAndSNCs_THolder callAndTopLevelConnectionsAndSNCs)

public void getCapabilities
  (org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

public void getConnectionsAndRouteDetails
  (java.lang.String callID,
   org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLRAName,
   java.lang.String sNPOrSNPPID,
   boolean mLSNPPLinkRequested,
   java.lang.String routeType,
   org.tmforum.mtnm.callSNC.SNCAndRouteList_THolder connectionAndRouteList)

public void getIntendedRoute
  (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
   boolean includeHigherOrderCCs,
   org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo,
   org.tmforum.mtnm.subnetworkConnection.Route_THolder route)

public void getMLSNPPLink
  (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] mLSNPPLinkName,
   boolean sNPLinkRequested,
   org.tmforum.mtnm.mLSNPPLink.MultiLayerSNPPLink_THolder theMLSNPPLink)
getMultiLayerSubnetwork

public void getMultiLayerSubnetwork(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] subnetName, org.tmforum.mtnm.multiLayerSubnetwork.MultiLayerSubnetwork_THolder subnetwork)

getRoute

public void getRoute(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName, boolean includeHigherOrderCCs, org.tmforum.mtnm.subnetworkConnection.Route_THolder route)

getRouteAndTopologicalLinks

public void getRouteAndTopologicalLinks(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName, org.tmforum.mtnm.subnetworkConnection.Route_THolder route, org.tmforum.mtnm.topologicalLink.TopologicalLinkList_THolder topologicalLinkList)

getSNC

public void getSNC(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName, org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder snc)

getSNCsByUserLabel

public void getSNCsByUserLabel(java.lang.String userLabel, org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sncList)
getTPGroupingRelationships

```java
public void getTPGroupingRelationships( NameAndStringValue_T[] tpName,
                                      int howMany,
                                      NamingAttributesList_THolder nameList,
                                      NamingAttributesIterator_IHolder nameIt)
```

getTPPool

```java
public void getTPPool( NameAndStringValue_T[] tPPoolName,
                       TerminationPoint_THolder tPPool,
                       IntHolder numberOfMembers,
                       IntHolder numberOfIdleMembers,
                       StringHolder descriptionOfUse)
```

getTopologicalLink

```java
public void getTopologicalLink( NameAndStringValue_T[] topoLinkName,
                                TopologicalLink_THolder topoLink)
```

modifyCall

```java
public void modifyCall( NameAndStringValue_T[] callName,
                        CallModifyData_T callModifyData,
                        Call_THolder modifiedCall)
```

modifyDiversityAndCorouting

public void modifyDiversityAndCorouting
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
     org.tmforum.mtnm.callSNC.Diversity_T callDiversity,
     org.tmforum.mtnm.callSNC.RouteGroupInfo_T[] routeGroupInfoList,
     boolean connectionRouteReArrangementAllowed,
     java.lang.String routeGroupsNumber,
     org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo,
     org.tmforum.mtnm.callSNC.CallAndTopLevelConnections_THolder callAndTopLevelConnections)

modifySNC

public void modifySNC
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
     java.lang.String routeId,
     org.tmforum.mtnm.subnetworkConnection.SNCModifyData_T SNCModifyData,
     org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
     org.tmforum.mtnm.subnetworkConnection.ProtectionEffort_T tolerableImpactEffort,
     org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
     org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
     org.tmforum.mtnm.subnetworkConnection.SubnetworkConnection_THolder newSNC,
     org.omg.CORBA.StringHolder errorReason)

modifyTPPool

public void modifyTPPool
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tPPoolName,
     org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] containedMembers,
     java.lang.String actionType,
     org.tmforum.mtnm.terminationPoint.TerminationPoint_THolder modifiedTPPool)

releaseCall

public void releaseCall
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
     org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
     org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder remainingSNCs,
     org.omg.CORBA.StringHolder errorReason)
removeConnections

```java
public void removeConnections(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] callName,
                            org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] connectionNamesList,
                            org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                            org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder sNCsNotDeleted,
                            org.omg.CORBA.StringHolder errorReason)
```

removeRoute

```java
public void removeRoute(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                        java.lang.String routeId,
                        org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                        org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
```

setAdditionalInfo

```java
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                               org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
```

setIntendedRoute

```java
public void setIntendedRoute(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                              java.lang.String routeId,
                              org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)
```

setNBI

```java
public void setNBI(NBIImp nbi)
```

*Sets the nBI.

**Parameters:**

- nbi - the new nBI*
setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                            java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)

setRoutesAdminState

public void setRoutesAdminState(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                                 org.tmforum.mtnm.subnetworkConnection.RouteNameAndAdminStateList_THolder routeNameAndAdminStateList,
                                 org.tmforum.mtnm.subnetworkConnection.SNCState_THolder sncState)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)

swapSNC

public void swapSNC(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] nameOfSNCtoBeDeactivated,
                    org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] nameOfSNCtoBeActivated,
                    org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                    org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                    org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                    org.tmforum.mtnm.subnetworkConnection.SNCState_THolder stateOfActivatedSNC,
                    org.omg.CORBA.StringHolder errorReason)
switchRoute

public void switchRoute(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] sncName,
                        java.lang.String routeId,
                        org.tmforum.mtnm.subnetworkConnection.GradesOfImpact_T tolerableImpact,
                        org.tmforum.mtnm.multiLayerSubnetwork.EMSFreedomLevel_T emsFreedomLevel,
                        org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
                        org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo,
                        org.tmforum.mtnm.subnetworkConnection.SNCState_THolder sncState,
                        org.omg.CORBA.StringHolder errorReason)

com.ericsson.poaImp

Class NamingAttributesIterator_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IPOA
         +--com.ericsson.poaImp.NamingAttributesIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IOperations

< Constructors > < Methods >

public class NamingAttributesIterator_IPOAImp extends org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IPOA
The Class NamingAttributesIterator_IPOAImp.

Constructors

NamingAttributesIterator_IPOAImp

public NamingAttributesIterator_IPOAImp(java.util.ArrayList list,
                                         NBIImp nbi,
                                         int size)

   Instantiates a new naming attributes iterator_ ipoa imp.

Parameters:
   list - the list
   nbi - the nbi
   size - the size
Methods

**destroy**

public void **destroy**()

---

**getLength**

public int **getLength**()

---

**next_n**

public boolean **next_n**(int howMany, org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList)

---

com.ericsson.poaImp

**Class NmsSession_IPOAImp**

java.lang.Object

---

org.omg.PortableServer.Servant

---

org.tmforum.mtnm.nmsSession.NmsSession_IPOA

---

com.ericsson.poaImp.NmsSession_IPOAImp

All Implemented Interfaces:

org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.nmsSession.NmsSession_IOperations

< Constructors > < Methods >

public class **NmsSession_IPOAImp** extends org.tmforum.mtnm.nmsSession.NmsSession_IPOA

The Class NmsSession_IPOAImp.

Constructors

**NmsSession_IPOAImp**

public **NmsSession_IPOAImp**()
Methods

alarmLossOccurred

public void alarmLossOccurred(java.lang.String startTime, java.lang.String notificationId)

associatedSession

public org.tmforum.mtnm.session.Session_I associatedSession()

dendSession

public void endSession()

eventLossCleared

public void eventLossCleared(java.lang.String endTime)

eventLossOccurred

public void eventLossOccurred(java.lang.String startTime, java.lang.String notificationId)

ping

public void ping()
com.ericsson.poaImp

Class PMDataIterator_IPOAImp

java.lang.Object
   |-- org.omg.PortableServer.Servant
      |-- org.tmforum.mtnm.performance.PMDataIterator_IPOA
         |-- com.ericsson.poaImp.PMDataIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.performance.PMDataIterator_IOperations

< Constructors > < Methods >

public class PMDataIterator_IPOAImp
extends org.tmforum.mtnm.performance.PMDataIterator_IPOA

The Class PMDataIterator_IPOAImp.

Constructors

PMDataIterator_IPOAImp

public PMDataIterator_IPOAImp()  

Methods

destroy

public void destroy()  

getLength

public int getLength()  

next_n

public boolean next_n(int howMany,
                        org.tmforum.mtnm.performance.PMDataList_THolder
                        pmDataList)
**com.ericsson.poaImp**

**Class PMPIterator_IPOAImp**

```java
class PMPIterator_IPOAImp
```

extends `org.tmforum.mtnm.performance.PMPIterator_IPOA`

All Implemented Interfaces:
- `org.omg.CORBA.portable.InvokeHandler`
- `org.tmforum.mtnm.performance.PMPIterator_IOperations`

### Constructors

**PMPIterator_IPOAImp**

```java
public PMPIterator_IPOAImp()
```

### Methods

**destroy**

```java
public void destroy()
```

**getLength**

```java
public int getLength()
```

**next_n**

```java
public boolean next_n(int howMany,
                      org.tmforum.mtnm.performance.PMPList_THolder pmpList)
```
com.ericsson.poaImp

Class PerformanceManagementMgr_IPOAImp

java.lang.Object
    |---org.omg.PortableServer.Servant
    |   |---org.tmforum.mtnm.performance.PerformanceManagementMgr_IPOA
    |   |---com.ericsson.poaImp.PerformanceManagementMgr_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
    org.tmforum.mtnm.performance.PerformanceManagementMgr_IOperations

< Constructors > < Methods >

public class PerformanceManagementMgr_IPOAImp
extends org.tmforum.mtnm.performance.PerformanceManagementMgr_IPOA

The Class PerformanceManagementMgr_IPOAImp.

Constructors

PerformanceManagementMgr_IPOAImp

public PerformanceManagementMgr_IPOAImp()

Methods

clearPMDData

public void clearPMData
   /org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
    org.tmforum.mtnm.performance.PMTPSelectList_THolder failedTPSelectList)
createTCAParameterProfile

public void createTCAParameterProfile
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName,
short layerRate,
java.lang.String userLabel,
boolean forceUniqueness,
java.lang.String owner,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] additionalInfo,
org.tmforum.mtnm.performance.TCAParameter_T[] listOfTCAParameter,
org.tmforum.mtnm.performance.TCAParameterProfile_THolder tcaParameterProfile)

deleteTCAParameterProfile

public void deleteTCAParameterProfile
(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tcaParameterProfileName)

disablePMData

public void disablePMData
(org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
org.tmforum.mtnm.performance.PMTPSelectList_THolder failedTPSelectList)

disableTCA

public void disableTCA
(org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
org.tmforum.mtnm.performance.PMTPSelectList_THolder failedTPSelectList)

enablePMData

public void enablePMData
(org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
org.tmforum.mtnm.performance.PMTPSelectList_THolder failedTPSelectList)
enableTCA

```java
public void enableTCA
    (org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
     org.tmforum.mtnm.performance.PMTPSelectList_THolder failedTPSelectList)
```

getAllCurrentPMData

```java
public void getAllCurrentPMData
    (org.tmforum.mtnm.performance.PMTPSelect_T[] pmTPSelectList,
     java.lang.String[] pmParameters,
     int howMany,
     org.tmforum.mtnm.performance.PMDDataList_THolder pmDataList,
     org.tmforum.mtnm.performance.PMDDataIterator_IHolder pmIt)
```

getAllPMPNames

```java
public void getAllPMPNames
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpOrMeName,
     int howMany,
     org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
     org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```

getAllPMPs

```java
public void getAllPMPs
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpOrMeName,
     int howMany,
     org.tmforum.mtnm.performance.PMPList_THolder pmpList,
     org.tmforum.mtnm.performance.PMPIterator_IHolder pmpIt)
```

getAllTCAParameterProfileNames

```java
public void getAllTCAParameterProfileNames
    (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
     int howMany,
     org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder tcaParameterProfileNames,
     org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```
getAllTCAParameterProfiles

public void getAllTCAParameterProfiles (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName, int howMany, org.tmforum.mtnm.performance.TCAParameterProfileList_THolder tcaParameterProfileList, org.tmforum.mtnm.performance.TCAParameterProfileIterator_IHolder tcaParameterProfileIt)

getCapabilities

public void getCapabilities (org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getHistoryPMData


getHoldingTime

public void getHoldingTime (org.tmforum.mtnm.performance.HoldingTime_THolder holdingTime)

getMEPMcapabilities

public void getMEPMcapabilities (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName, short layerRate, org.tmforum.mtnm.performance.PMParameterList_THolder pmParameterList)
getProfileAssociatedTPs

```java
public void getProfileAssociatedTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
groundName,
                 int howMany,
ground.tmforum.mtnm.globaldefs.NamingAttributesList_THolder tpNames,
ground.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)
```

getTCAPParameterProfile

```java
public void getTCAPParameterProfile(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
tcaParameterProfileName,
ground.tmforum.mtnm.performance.TCAParameterProfile_THolder tcaParameterProfile)
```

getTCATPParameter

```java
public void getTCATPParameter(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
short layerRate,
java.lang.String granularity,
ground.tmforum.mtnm.performance.TCAParameters_THolder tcaParameter)
```

getTPHistoryPMData

```java
public void getTPHistoryPMData(org.tmforum.mtnm.performance.PMTPSelect_T[]
pmTPSelectList,
java.lang.String[] pmParameters,
java.lang.String startTime,
java.lang.String endTime,
int howMany,
ground.tmforum.mtnm.performance.PMDataList_THolder pmDataList,
ground.tmforum.mtnm.performance.PMDataIterator_IHolder pmIt)
```

setAdditionalInfo

```java
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
objectName,
ground.tmforum.mtnm.globaldefs.NVSLList_THolder additionalInfo)
```

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                              java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)

setTCAPerParameterProfile

public void setTCAPerParameterProfile(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tcaParameterProfileName,
                                       org.tmforum.mtnm.performance.TCAParameter_T[] listOfTCAParameter,
                                       int howMany,
                                       org.tmforum.mtnm.performance.TCAParameterProfile_THolder tcaParameterProfile,
                                       org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder failedTPList,
                                       org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

setTCAPerParameterProfilePointer

public void setTCAPerParameterProfilePointer(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                                              org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] addTCAParameterProfile,
                                              org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] removeTCAParameterProfile)

setTCATPParameter

public void setTCATPParameter(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tpName,
                               org.tmforum.mtnm.performance.TCAParameters_THolder tcaParameters)
**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                          java.lang.String userLabel,
                          boolean enforceUniqueness)
```

---

**com.ericsson.poaImp**

**Class ProtectionGroupIterator_IPOAImp**

```java
java.lang.Object
  `+-org.omg.PortableServer.Servant
    `+-org.tmforum.mtnm.protection.ProtectionGroupIterator_IPOA
      `+-com.ericsson.poaImp.ProtectionGroupIterator_IPOAImp
```

**All Implemented Interfaces:**
- org.omg.CORBA.portable.InvokeHandler,
- org.tmforum.mtnm.protection.ProtectionGroupIterator_IOperations

< Constructors > < Methods >

Public class **ProtectionGroupIterator_IPOAImp**

extends org.tmforum.mtnm.protection.ProtectionGroupIterator_IPOA

The Class ProtectionGroupIterator_IPOAImp.

---

**Constructors**

**ProtectionGroupIterator_IPOAImp**

```java
public ProtectionGroupIterator_IPOAImp()
```

---

**Methods**

**destroy**

```java
public void destroy()
```

---

**getLength**

```java
public int getLength()
```
### next_n

```java
public boolean next_n(int howMany,
                      org.tmforum.mtnm.protection.ProtectionGroupList_THolder pgpList)
```

---

### com.ericsson.poaImp

**Class ProtectionMgr_IPOAImp**

```java
java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.protection.ProtectionMgr_IPOA
         +--com.ericsson.poaImp.ProtectionMgr_IPOAImp
```

**All Implemented Interfaces:**
- org.omg.CORBA.portable.InvokeHandler,
- org.tmforum.mtnm.protection.ProtectionMgr_IOperations

---

**< Constructors >**

```java
public class ProtectionMgr_IPOAImp
extends org.tmforum.mtnm.protection.ProtectionMgr_IPOA

The Class ProtectionMgr_IPOAImp.
```

---

### Constructors

**ProtectionMgr_IPOAImp**

```java
public ProtectionMgr_IPOAImp()
```

---

### Methods

**getAllEProtectionGroups**

```java
public void getAllEProtectionGroups(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName,
                                    int howMany,
                                    org.tmforum.mtnm.protection.EProtectionGroupList_THolder epgpList,
                                    org.tmforum.mtnm.protection.EProtectionGroupIterator_IHolder epgpIt)
```
getAllNUTTPNames

public void getAllNUTTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] pgName, 
                              int howMany, 
                              org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList, 
                              org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllPreemptibleTPNames

public void getAllPreemptibleTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] pgName, 
                                      int howMany, 
                                      org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList, 
                                      org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllProtectedTPNames

public void getAllProtectedTPNames(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] pgName, 
                                    int howMany, 
                                    org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList, 
                                    org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllProtectionGroups

public void getAllProtectionGroups(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] meName, 
                                   int howMany, 
                                   org.tmforum.mtnm.protection.ProtectionGroupList_THolder pgList, 
                                   org.tmforum.mtnm.protection.ProtectionGroupIterator_IHolder pgpIt)

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
getContainingPGNames
public void getContainingPGNames (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] pTPName,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder pgNameList)

getEProtectionGroup
public void getEProtectionGroup (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] ePGPname,
org.tmforum.mtnm.protection.EProtectionGroup_THolder eProtectionGroup)

getProtectionGroup
public void getProtectionGroup (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] pgName,
org.tmforum.mtnm.protection.ProtectionGroup_THolder protectionGroup)

performProtectionCommand
public void performProtectionCommand (org.tmforum.mtnm.protection.ProtectionCommand_T protectionCommand,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] reliableSinkCtpOrGroupName,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] fromTp,
org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] toTp,
org.tmforum.mtnm.protection.SwitchData_THolder switchData)

retrieveESwitchData
public void retrieveESwitchData (org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] ePGPName,
org.tmforum.mtnm.protection.ESwitchDataList_THolder eSwitchDataList)
**retrieveSwitchData**

```java
public void retrieveSwitchData(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] reliableSinkCtpOrGroupName,
                               org.tmforum.mtnm.protection.SwitchDataList_THolder switchData)
```

**setAdditionalInfo**

```java
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
                              objectName,
                              org.tmforum.mtnm.globaldefs.NVSLList_THolder additionalInfo)
```

**setNativeEMSName**

```java
public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
                              objectName,
                              java.lang.String nativeEMSName)
```

**setOwner**

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
                     objectName,
                     java.lang.String owner)
```

**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[]
                          objectName,
                          java.lang.String userLabel,
                          boolean enforceUniqueness)
```
Class SNCIterator_IPOAImp

java.lang.Object
   |--org.omg.PortableServer.Servant
      |--org.tmforum.mtnm.subnetworkConnection.SNCIterator_IPOA
         |--com.ericsson.poaImp.SNCIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.subnetworkConnection.SNCIterator_IOperations

Constructors

SNCIterator_IPOAImp

public SNCIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany,
                      org.tmforum.mtnm.subnetworkConnection.SubnetworkConnectionList_THolder
                      sncList)
**Class Session_IPOAImp**

The Class Session_IPOAImp.

### Constructors

**Session_IPOAImp**

```java
public Session_IPOAImp()
```

### Methods

**associatedSession**

```java
public org.tmforum.mtnm.session.Session_I associatedSession()
```

**endSession**

```java
public void endSession()
```

**ping**

```java
public void ping()
```
Constructors

SoftwareAndDataMgr_IPOAImpl

public SoftwareAndDataMgr_IPOAImpl()

Methods

abortMEBackup

public void abortMEBackup(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName)

backupME

public void backupME(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName)
getBackupList

public void getBackupList(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[][] managedElementNameList, int howMany, org.tmforum.mtnm.softwareAndDataManager.BackupIdList_THolder backupList, org.tmforum.mtnm.softwareAndDataManager.BackupIdIterator_IHolder backupIt)

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getMEBackupStatus

public void getMEBackupStatus(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] managedElementName, org.tmforum.mtnm.softwareAndDataManager.BackupStatus_THolder backupStatus)

setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName, java.lang.String owner)
**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                          java.lang.String userLabel,
                          boolean enforceUniqueness)
```

---

**com.ericsson.poalmp**

**Class SubnetworkIterator_IPOAImp**

java.lang.Object
  └--org.omg.PortableServer.Servant
    └--org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IPOA
      └--com.ericsson.poalmp.SubnetworkIterator_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IOperations

< Constructors > < Methods >

public class SubnetworkIterator_IPOAImp
extends org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkIterator_IPOA

The Class SubnetworkIterator_IPOAImp.

**Constructors**

**SubnetworkIterator_IPOAImp**

```java
public SubnetworkIterator_IPOAImp(java.util.ArrayList list,
                                 NBIImp nbi,
                                 int size)
```

Instantiates a new subnetwork iterator_ipoa_imp.

Parameters:
  - list - the list
  - nbi - the nbi
  - size - the size

**Methods**
destroy
public void destroy()

getLength
public int getLength()

next_n
public boolean next_n(int howMany,
org.tmforum.mtnm.multiLayerSubnetwork.SubnetworkList_THolder subnetworkList)

com.ericsson.poaImp

Class TCAParameterProfileIterator_IPOAImpl
java.lang.Object
    |--org.omg.PortableServer.Servant
        |--org.tmforum.mtnm.performance.TCAParameterProfileIterator_IPOA
            |--com.ericsson.poaImp.TCAParameterProfileIterator_IPOAImpl

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
    org.tmforum.mtnm.performance.TCAParameterProfileIterator_IOperations

< Constructors > < Methods >

public class TCAParameterProfileIterator_IPOAImpl
extends org.tmforum.mtnm.performance.TCAParameterProfileIterator_IPOA

The Class TCAParameterProfileIterator_IPOAImpl.

Constructors

TCAParameterProfileIterator_IPOAImpl
public TCAParameterProfileIterator_IPOAImpl()
**destroy**

```java
public void destroy()
```

---

**getLength**

```java
public int getLength()
```

---

**next_n**

```java
public boolean next_n(int howMany,
org.tmforum.mtnm.performance.TCAParameterProfileList_THolder
tcaParameterProfileList)
```

---

**com.ericsson.poaImp**

**Class TCProfileIterator_IPOAImp**

```java
java.lang.Object
|--org.omg.PortableServer.Servant
|   |--org.tmforum.mtnm.trafficConditioningProfile.TCProfileIterator_IPOA
|     |--com.ericsson.poaImp.TCProfileIterator_IPOAImp
```

**All Implemented Interfaces:**

- org.omg.CORBA.portable.InvokeHandler
- org.tmforum.mtnm.trafficConditioningProfile.TCProfileIterator_IOperations

---

< Constructors > < Methods >

**public class TCProfileIterator_IPOAImp**

extends org.tmforum.mtnm.trafficConditioningProfile.TCProfileIterator_IPOA

The Class TCProfileIterator_IPOAImp.

---

**Constructors**

**TCProfileIterator_IPOAImp**

```java
public TCProfileIterator_IPOAImp()
```

---

**Methods**
destroy
public void destroy()

getLength
public int getLength()

next_n
public boolean next_n(int howMany,
org.tmforum.mtnm.trafficConditioningProfile.TCProfileList_THolder tcProfileList)

com.ericsson.poalmp

Class TCProfileMgr_IPOAImp

down java.lang.Object
        |   +--org.omg.PortableServer.Servant
              |   +--org.tmforum.mtnm.trafficConditioningProfile.TCProfileMgr_IPOA
                       |   +--com.ericsson.poalmp.TCProfileMgr_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.trafficConditioningProfile.TCProfileMgr_IOperations

< Constructors >< Methods >

public class TCProfileMgr_IPOAImp
extends org.tmforum.mtnm.trafficConditioningProfile.TCProfileMgr_IPOA

The Class TCProfileMgr_IPOAImp.

Constructors

TCPProfileMgr_IPOAImp
public     TCPProfileMgr_IPOAImp()
createTCProfile

public void createTCProfile(org.tmforum.mtnm.trafficConditioningProfile.TCProfileCreateData_T newTCProfileCreateData,
org.tmforum.mtnm.trafficConditioningProfile.TCProfile_THolder newTCProfile)

deleteTCProfile

public void deleteTCProfile(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tcProfileName)

getAllTCProfiles

public void getAllTCProfiles(int howMany,
org.tmforum.mtnm.trafficConditioningProfile.TCProfileList_THolder tcProfileList,
org.tmforum.mtnm.trafficConditioningProfile.TCProfileIterator_IHolder tcProfileIt)

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getTCProfile

public void getTCProfile(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tcProfileName,
org.tmforum.mtnm.trafficConditioningProfile.TCProfile_THolder tcProfile)

getTCProfileAssociatedTPs

public void getTCProfileAssociatedTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tcProfileName,
int howMany,
org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
modifyTCProfile

```java
public void modifyTCProfile(String tcProfileName, TrafficConditioningProfile.TCProfileCreateData_tcProfileModifyData, TrafficConditioningProfile.TCProfile_tcProfileModifyData, org.omg.CORBA.StringHolder errorReason)
```

setAdditionalInfo

```java
public void setAdditionalInfo(String objectName, NVSLIST_THolder additionalInfo)
```

setNativeEMSName

```java
public void setNativeEMSName(String objectName, String nativeEMSName)
```

setOwner

```java
public void setOwner(String objectName, String owner)
```

setUserLabel

```java
public void setUserLabel(String objectName, String userLabel, boolean enforceUniqueness)
```
com.ericsson.poailmp

Class TerminationPointIterator_IPOAImp

data type Object

|--org.omg.PortableServer.Servant

|--org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IPOA

|--com.ericsson.poailmp.TerminationPointIterator_IPOAImp

All Implemented Interfaces:

org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IOperations

< Constructors > < Methods >

public class TerminationPointIterator_IPOAImp
extends org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IPOA

The Class TerminationPointIterator_IPOAImp.

Constructors

TerminationPointIterator_IPOAImp

public TerminationPointIterator_IPOAImp(java.util.ArrayList list,
NBIImp nbi,
int size)

Instantiates a new termination point iterator_ipoa imp.

Parameters:

list - the list
nbi - the nbi
size - the size

Methods

destroy

public void destroy()
public boolean next_n(int howMany, org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList)

com.ericsson.poalmp

Class TopologicalLinkIterator_IPOAImp

java.lang.Object
    |-- org.omg.PortableServer.Servant
    |    |-- org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IPOA
    |    |-- com.ericsson.poalmp.TopologicalLinkIterator_IPOAImp

All Implemented Interfaces:
    org.omg.CORBA.portable.InvokeHandler,
    org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IOperations

< Constructors > < Methods >

public class TopologicalLinkIterator_IPOAImp
extends org.tmforum.mtnm.topologicalLink.TopologicalLinkIterator_IPOA

The Class TopologicalLinkIterator_IPOAImp.

Constructors

TopologicalLinkIterator_IPOAImp

public TopologicalLinkIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()
public boolean next_n(int howMany, org.tmforum.mtnm.topologicalLink.TopologicalLinkList_THolder topoLinkList)

com.ericsson.poalmp

Class TrafficDescriptorIterator_IPOAImp

java.lang.Object
  |--org.omg.PortableServer.Servant
  |    |--org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorIterator_IPOA
  |       |--com.ericsson.poalmp.TrafficDescriptorIterator_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorIterator_IOperations

< Constructors > < Methods >

public class TrafficDescriptorIterator_IPOAImp
  extends org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorIterator_IPOA

The Class TrafficDescriptorIterator_IPOAImp.

Constructors

TrafficDescriptorIterator_IPOAImp

public TrafficDescriptorIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()
public boolean next_n(int howMany,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorList_THolder
trafficDescList)

com.ericsson.poalmp

Class TrafficDescriptorMgr_IPOAImp

does things for creating and managing traffic descriptors.

java.lang.Object
   |--org.omg.PortableServer.Servant
      |--org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorMgr_IPOA
         |--com.ericsson.poalmp.TrafficDescriptorMgr_IPOAImp

All Implemented Interfaces: org.omg.CORBA.portable.InvokeHandler,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorMgr_IOperations

< Constructors > < Methods >

public class TrafficDescriptorMgr_IPOAImp
extends org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorMgr_IPOA

The Class TrafficDescriptorMgr_IPOAImp.

Constructors

TrafficDescriptorMgr_IPOAImp

public TrafficDescriptorMgr_IPOAImp()

Methods

createTrafficDescriptor

public void createTrafficDescriptor(org.tmforum.mtnm.trafficDescriptor.TDCreateData_T
newTDCreateData,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptor_THolder
newTrafficDescriptor)
deleteTrafficDescriptor

public void deleteTrafficDescriptor(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] descriptorName)

getAllTrafficDescriptorNames

public void getAllTrafficDescriptorNames(int howMany,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllTrafficDescriptors

public void getAllTrafficDescriptors(int howMany,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorList_THolder trafficDescList,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptorIterator_IHolder trafficDescIt)

getAssociatedCTPs

public void getAssociatedCTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] trafficDescriptorName,
int howMany,
org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)

getCapabilities

public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)

getTrafficDescriptor

public void getTrafficDescriptor(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tdName,
org.tmforum.mtnm.trafficDescriptor.TrafficDescriptor_THolder td)
setAdditionalInfo

public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
org.tmforum.mtnm.globaldefs.NVSList_THolder additionalInfo)

setNativeEMSName

public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
java.lang.String nativeEMSName)

setOwner

public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
java.lang.String owner)

setUserLabel

public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
java.lang.String userLabel,
boolean enforceUniqueness)

com.ericsson.poaImp

Class TransmissionDescriptorIterator_IPOAImp

java.lang.Object
   |---org.omg.PortableServer.Servant
      |   |---org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorIterator_IPOA
         |   |---com.ericsson.poaImp.TransmissionDescriptorIterator_IPOAImp

All Implemented Interfaces:
   org.omg.CORBA.portable.InvokeHandler,
   org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorIterator_IOperations

< Constructors > < Methods >
public class TransmissionDescriptorIterator_IPOAImp extends org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorIterator_IPOA

The Class TransmissionDescriptorIterator_IPOAImp.

Constructors

TransmissionDescriptorIterator_IPOAImp

public TransmissionDescriptorIterator_IPOAImp()

Methods

destroy

public void destroy()

getLength

public int getLength()

next_n

public boolean next_n(int howMany, org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorList_THolder transmissionDescList)

com.ericsson.poaImp

Class TransmissionDescriptorMgr_IPOAImp

java.lang.Object
  |----org.omg.PortableServer.Servant
  |     |----org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorMgr_IPOA
  |     |     |----com.ericsson.poaImp.TransmissionDescriptorMgr_IPOAImp

All Implemented Interfaces:
  org.omg.CORBA.portable.InvokeHandler,
  org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorMgr_IOperations

< Constructors > < Methods >
The Class TransmissionDescriptorMgr_IPOAImp.

Constructors

TransmissionDescriptorMgr_IPOAImp

public TransmissionDescriptorMgr_IPOAImp()

Methods

createTransmissionDescriptor

public void createTransmissionDescriptor(org.tmforum.mtnm.transmissionDescriptor.TMDCreateData_T newTMDCreateData,
org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptor_THolder newTransmissionDescriptor)

deleteTransmissionDescriptor

public void deleteTransmissionDescriptor(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] transmissionDescriptorName)

getAllTransmissionDescriptorNames

public void getAllTransmissionDescriptorNames(int howMany,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder nameList,
org.tmforum.mtnm.globaldefs.NamingAttributesIterator_IHolder nameIt)

getAllTransmissionDescriptors

public void getAllTransmissionDescriptors(int howMany,
org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorList_THolder transmissionDescList,
org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptorIterator_IHolder transmissionDescIt)
getAssociatedTPs

```java
public void getAssociatedTPs(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] transmissionDescriptorName,
int howMany,
org.tmforum.mtnm.terminationPoint.TerminationPointList_THolder tpList,
org.tmforum.mtnm.terminationPoint.TerminationPointIterator_IHolder tpIt)
```

getCapabilities

```java
public void getCapabilities(org.tmforum.mtnm.common.CapabilityList_THolder capabilities)
```

getTransmissionDescriptor

```java
public void getTransmissionDescriptor(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tmdName,
org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptor_THolder tmd)
```

modifyTransmissionDescriptor

```java
public void modifyTransmissionDescriptor(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tmdName,
org.tmforum.mtnm.transmissionDescriptor.TMDModifyData_T tmdModifyData,
org.tmforum.mtnm.subnetworkConnection.TPDataList_THolder tpsToModify,
org.tmforum.mtnm.transmissionDescriptor.TransmissionDescriptor_THolder modifiedTransmissionDescriptor,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder failedMEList,
org.tmforum.mtnm.globaldefs.NamingAttributesList_THolder failedTPsMFDsList,
org.omg.CORBA.StringHolder errorReason)
```

setAdditionalInfo

```java
public void setAdditionalInfo(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
additionalInfo)
```
**setNativeEMSName**

```java
public void setNativeEMSName(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                            java.lang.String nativeEMSName)
```

**setOwner**

```java
public void setOwner(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                     java.lang.String owner)
```

**setTMDAssociation**

```java
public void setTMDAssociation(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] tmdName,
                               org.tmforum.mtnm.transmissionDescriptor.TPorMFDorFDFr_THolder tPorMFDorFDFr)
```

**setUserLabel**

```java
public void setUserLabel(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                         java.lang.String userLabel,
                         boolean enforceUniqueness)
```

**validateTMDAssignmentToObject**

```java
public void validateTMDAssignmentToObject(org.tmforum.mtnm.globaldefs.NameAndStringValue_T[] objectName,
                                          org.omg.CORBA.StringHolder objectAssignmentState,
                                          org.tmforum.mtnm.transmissionParameters.LayeredParameterList_THolder transmissionParams,
                                          org.tmforum.mtnm.globaldefs.NVSList_THolder additionalTPInfo)
```
com.ericsson.poalmp

Class Version_IPOAImp

java.lang.Object
   +--org.omg.PortableServer.Servant
      +--org.tmforum.mtnm.mtnmVersion.Version_IPOA
         +--com.ericsson.poalmp.Version_IPOAImp

All Implemented Interfaces:

< Constructors > < Methods >

public class Version_IPOAImp
extends org.tmforum.mtnm.mtnmVersion.Version_IPOA

This class implements part of the TMF814 SS 3.5.

Constructors

Version_IPOAImp

public Version_IPOAImp()

Methods

getVersion

public java.lang.String getVersion()

   Returns the version of the Solution Set of the TMF idl-files upon which this interface is build.
   Returns:
      The version number of the Solution Set of TMF814.
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>221</td>
</tr>
<tr>
<td></td>
<td>241</td>
</tr>
<tr>
<td></td>
<td>208</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>295</td>
</tr>
<tr>
<td></td>
<td>295</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>311</td>
</tr>
<tr>
<td></td>
<td>241</td>
</tr>
<tr>
<td></td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>251</td>
</tr>
<tr>
<td></td>
<td>251</td>
</tr>
<tr>
<td></td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>166</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>241</td>
</tr>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>216</td>
</tr>
<tr>
<td></td>
<td>216</td>
</tr>
<tr>
<td></td>
<td>217</td>
</tr>
<tr>
<td></td>
<td>217</td>
</tr>
<tr>
<td></td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>218</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>318</td>
</tr>
<tr>
<td></td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>242</td>
</tr>
<tr>
<td></td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>341</td>
</tr>
<tr>
<td></td>
<td>345</td>
</tr>
<tr>
<td></td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>239</td>
</tr>
<tr>
<td></td>
<td>240</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>234</td>
</tr>
<tr>
<td></td>
<td>234</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>193</td>
</tr>
<tr>
<td></td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>241</td>
</tr>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>286</td>
</tr>
<tr>
<td></td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>318</td>
</tr>
<tr>
<td></td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>242</td>
</tr>
<tr>
<td></td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>341</td>
</tr>
<tr>
<td></td>
<td>345</td>
</tr>
<tr>
<td></td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>235</td>
</tr>
<tr>
<td></td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>238</td>
</tr>
<tr>
<td></td>
<td>239</td>
</tr>
<tr>
<td></td>
<td>240</td>
</tr>
</tbody>
</table>
deactivateAndDeleteEDFr ... 264
deactivateAndDeleteSNC ... 297
deactivateSNC ... 297
deassignASAP ... 242
deassignSignallingController ... 278
deAssociateCPTPsFromFlowDomain ... 263
deAssociateMFDsFromFlowDomain ... 263
debug ... 103
debug ... 221
deleteAlarm ... 22
deleteAlarm ... 70
deleteAlarms ... 122
deleteASAP ... 242
deleteCTP ... 22
deleteCTP ... 71
deleteCTP ... 128
deleteEms ... 22
deleteEms ... 71
deleteEms ... 133
deleteFlowDomain ... 263
deleteFTP ... 264
deleteGTP ... 286
deleteME ... 23
deleteME ... 71
deleteME ... 141
deleteMFD ... 264
deleteMLSN ... 23
deleteMLSN ... 71
deleteMLSN ... 145
deletePTP ... 23
deletePTP ... 72
deletePTP ... 148
deleteSNC ... 297
deleteTCA ... 23
deleteTCA ... 72
deleteTCAP-parameterProfile ... 318
deleteTCAs ... 155
deleteTCP-profile ... 336
deleteTL ... 24
deleteTL ... 72
deleteTL ... 158
deleteTopologicalLink ... 242
deleteTPPool ... 297
deleteTrafficDescriptor ... 342
deleteTransmissionDescriptor ... 345
destroy ... 233
destroy ... 234
destroy ... 235
destroy ... 236
destroy ... 237
destroy ... 240
destroy ... 249
destroy ... 257
destroy ... 259
destroy ... 260
destroy ... 261
destroy ... 270
destroy ... 274
destroy ... 275
destroy ... 276
destroy ... 277
destroy ... 285
destroy ... 313
destroy ... 315
destroy ... 316
destroy ... 323
destroy ... 328
destroy ... 333
destroy ... 334
destroy ... 335
destroy ... 338
destroy ... 339
destroy ... 340
destroy ... 344
destroyGCT ... 271
disablePMData ... 318
disableTCA ... 318
disableSignalling ... 278
disconnect ... 36
disconnect ... 72
DBI ... 70
Debugger ... 61
Debugger ... 61
DebugInterface ... 101
DynamicLoader ... 64
DynamicLoader ... 64
getActive ... 103
getActive ... 221
getActiveAlarms ... 25
getActiveAlarms ... 73
getActiveMaintenanceOperations ... 283
getActiveTCAs ... 25
getActiveTCAs ... 73
getAdditionalInfo ... 25
getAdditionalInfo ... 73
getAdditionalInfo ... 103
getAdditionalInfo ... 136
getAdditionalInfo ... 221
getAlarm ... 25
getAlarm ... 73
getAlarm ... 122
getAlarms ... 104
getAlarms ... 122
getAlarms ... 222
getAlarmTypes ... 6
getAllActiveAlarms ... 26
getAllActiveAlarms ... 74
getAllActiveAlarms ... 123
getAllActiveAlarms ... 286
getAllActiveAlarmsFiltered ... 26
getAllActiveAlarmsFiltered ... 74
getAllActiveTCAs ... 26
getAllActiveTCAs ... 74
getAllActiveTCAs ... 156
getAllActiveTCAsFiltered ... 26
getAllActiveTCAsFiltered ... 74
getAllAlarmParameterNames ... 27
getAllAlarmParameterNames ... 75
getAllAlarmParameters ... 123
getAllAlarms ... 104
getAllAlarms ... 222
getAllASAPNames ... 243
getAllASAPs ... 243
getAllAssignableCPTPs ... 264
getAllAssignedCPTPs ... 265
getAllAssociatedMFDs ... 265
getAllCallIdsWithSNPPOrTNAName ... 298
getAllCallIdsWithTP ... 298
getAllCallsAndTopLevelConnections ... 298
getAllCallsAndTopLevelConnectionsAndSNCs ... 299
getAllCallsAndTopLevelConnectionsAndSNCsWithME ... 299
getAllCallsAndTopLevelConnectionsAndSNCsWithTP ... 299
getAllCallsAndTopLevelConnectionsWithME ... 299
getAllCPTPs ... 265
getAllCrossConnections ... 286
getAllCurrentPMData ... 319
getAllEdgeMLSNPPLinks ... 300
getAllEdgePointNames ... 300
getAllEdgePoints ... 300
getAllEMSAndMEActiveAlarms ... 243
getAllEMSAndMENotifiedActiveAlarms ... 243
getAllEMSSystemActiveAlarms ... 244
getAllEMSSystemUnacknowledgedActiveAlarms ... 244
getAllEProtectionGroups ... 324
getAllEquipment ... 253

G

disablePMData ... 318
disableTCA ... 278
disableSignalling ... 278
disconnect ... 36
disconnect ... 72
DBI ... 70
Debugger ... 61
Debugger ... 61
DebugInterface ... 101
DynamicLoader ... 64
DynamicLoader ... 64
getActive ... 103
getActive ... 221
getActiveAlarms ... 25
getActiveAlarms ... 73
getActiveMaintenanceOperations ... 283
getActiveTCAs ... 25
getActiveTCAs ... 73
getAdditionalInfo ... 25
getAdditionalInfo ... 73
getAdditionalInfo ... 103
getAdditionalInfo ... 136
getAdditionalInfo ... 221
getAlarm ... 25
getAlarm ... 73
getAlarm ... 122
getAlarms ... 104
getAlarms ... 122
getAlarms ... 222
getAlarmTypes ... 6
getAllActiveAlarms ... 26
getAllActiveAlarms ... 74
getAllActiveAlarms ... 123
getAllActiveAlarms ... 286
getAllActiveAlarmsFiltered ... 26
getAllActiveAlarmsFiltered ... 74
getAllActiveTCAs ... 26
getAllActiveTCAs ... 74
getAllActiveTCAs ... 156
getAllActiveTCAsFiltered ... 26
getAllActiveTCAsFiltered ... 74
getAllAlarmParameterNames ... 27
getAllAlarmParameterNames ... 75
getAllAlarmParameters ... 123
getAllAlarms ... 104
getAllAlarms ... 222
getAllASAPNames ... 243
getAllASAPs ... 243
getAllAssignableCPTPs ... 264
getAllAssignedCPTPs ... 265
getAllAssociatedMFDs ... 265
getAllCallIdsWithSNPPOrTNAName ... 298
getAllCallIdsWithTP ... 298
getAllCallsAndTopLevelConnections ... 298
getAllCallsAndTopLevelConnectionsAndSNCs ... 299
getAllCallsAndTopLevelConnectionsAndSNCsWithME ... 299
getAllCallsAndTopLevelConnectionsAndSNCsWithTP ... 299
getAllCallsAndTopLevelConnectionsWithME ... 299
getAllCPTPs ... 265
getAllCrossConnections ... 286
getAllCurrentPMData ... 319
getAllEdgeMLSNPPLinks ... 300
getAllEdgePointNames ... 300
getAllEdgePoints ... 300
getAllEMSAndMEActiveAlarms ... 243
getAllEMSAndMENotifiedActiveAlarms ... 243
getAllEMSSystemActiveAlarms ... 244
getAllEMSSystemUnacknowledgedActiveAlarms ... 244
getAllEProtectionGroups ... 324
getAllEquipment ... 253
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>performMaintenanceOperation</td>
<td>283</td>
</tr>
<tr>
<td>performProtectionCommand</td>
<td>326</td>
</tr>
<tr>
<td>ping</td>
<td>252</td>
</tr>
<tr>
<td>ping</td>
<td>314</td>
</tr>
<tr>
<td>ping</td>
<td>329</td>
</tr>
<tr>
<td>print</td>
<td>175</td>
</tr>
<tr>
<td>printHashMap</td>
<td>139</td>
</tr>
<tr>
<td>printProperty</td>
<td>218</td>
</tr>
<tr>
<td>provisionEquipment</td>
<td>255</td>
</tr>
<tr>
<td>ParseStatusPanel</td>
<td>174</td>
</tr>
<tr>
<td>ParseStatusPanel</td>
<td>175</td>
</tr>
<tr>
<td>PerformanceManagementMgr_IPOAImp</td>
<td>317</td>
</tr>
<tr>
<td>PerformanceManagementMgr_IPOAImp</td>
<td>317</td>
</tr>
<tr>
<td>PMDataIterator_IPOAImp</td>
<td>315</td>
</tr>
<tr>
<td>PMDataIterator_IPOAImp</td>
<td>315</td>
</tr>
<tr>
<td>PMPIterator_IPOAImp</td>
<td>316</td>
</tr>
<tr>
<td>PMPIterator_IPOAImp</td>
<td>316</td>
</tr>
<tr>
<td>ProtectionGroupIterator_IPOAImp</td>
<td>323</td>
</tr>
<tr>
<td>ProtectionGroupIterator_IPOAImp</td>
<td>323</td>
</tr>
<tr>
<td>ProtectionMgr_IPOAImp</td>
<td>324</td>
</tr>
<tr>
<td>ProtectionMgr_IPOAImp</td>
<td>324</td>
</tr>
<tr>
<td>rackCall</td>
<td>309</td>
</tr>
<tr>
<td>removeConnections</td>
<td>310</td>
</tr>
<tr>
<td>removeCurrentNode</td>
<td>19</td>
</tr>
<tr>
<td>removeHardParentheses</td>
<td>68</td>
</tr>
<tr>
<td>removeParentheses</td>
<td>68</td>
</tr>
<tr>
<td>removeRoute</td>
<td>310</td>
</tr>
<tr>
<td>repaintStatus</td>
<td>176</td>
</tr>
<tr>
<td>reset</td>
<td>37</td>
</tr>
<tr>
<td>reset</td>
<td>83</td>
</tr>
<tr>
<td>reset</td>
<td>193</td>
</tr>
<tr>
<td>reset</td>
<td>197</td>
</tr>
<tr>
<td>reSetConfig</td>
<td>9</td>
</tr>
<tr>
<td>resetDB</td>
<td>140</td>
</tr>
<tr>
<td>retrieveESwitchData</td>
<td>326</td>
</tr>
<tr>
<td>retrieveSwitchData</td>
<td>327</td>
</tr>
<tr>
<td>run</td>
<td>60</td>
</tr>
<tr>
<td>run</td>
<td>163</td>
</tr>
<tr>
<td>run</td>
<td>164</td>
</tr>
<tr>
<td>run</td>
<td>227</td>
</tr>
<tr>
<td>RTAMAlarm</td>
<td>205</td>
</tr>
<tr>
<td>RTAMAlarm</td>
<td>206</td>
</tr>
<tr>
<td>RTAMController</td>
<td>13</td>
</tr>
<tr>
<td>RTAMController</td>
<td>13</td>
</tr>
<tr>
<td>RTAMFilter</td>
<td>208</td>
</tr>
<tr>
<td>RTAMFilter</td>
<td>208</td>
</tr>
<tr>
<td>RTAMTCA</td>
<td>209</td>
</tr>
<tr>
<td>RTAMTCA</td>
<td>210</td>
</tr>
<tr>
<td>RTAMView</td>
<td>212</td>
</tr>
<tr>
<td>RTAMView</td>
<td>212</td>
</tr>
<tr>
<td>sendAlarm</td>
<td>112</td>
</tr>
<tr>
<td>sendAlarm</td>
<td>227</td>
</tr>
<tr>
<td>sendAttributeChanged</td>
<td>113</td>
</tr>
<tr>
<td>sendAttributeChanged</td>
<td>227</td>
</tr>
<tr>
<td>sendHeartBeat</td>
<td>113</td>
</tr>
<tr>
<td>sendHeartBeat</td>
<td>227</td>
</tr>
<tr>
<td>sendStateChanged</td>
<td>114</td>
</tr>
<tr>
<td>sendStateChanged</td>
<td>228</td>
</tr>
<tr>
<td>setActive</td>
<td>114</td>
</tr>
<tr>
<td>setActive</td>
<td>228</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>38</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>239</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>247</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>256</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>269</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>272</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>279</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>281</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>283</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>293</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>310</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>321</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>327</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>331</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>337</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>343</td>
</tr>
<tr>
<td>setAdditionalInfo</td>
<td>346</td>
</tr>
<tr>
<td>setAlarmReportingOff</td>
<td>256</td>
</tr>
<tr>
<td>setAlarmReportingOn</td>
<td>256</td>
</tr>
<tr>
<td>setClient</td>
<td>228</td>
</tr>
<tr>
<td>setColumnNames</td>
<td>207</td>
</tr>
<tr>
<td>setColumnNames</td>
<td>211</td>
</tr>
<tr>
<td>setCrossConnection</td>
<td>40</td>
</tr>
<tr>
<td>setCrossConnection</td>
<td>85</td>
</tr>
<tr>
<td>setCTP</td>
<td>39</td>
</tr>
<tr>
<td>setCTP</td>
<td>84</td>
</tr>
<tr>
<td>setCtpModel</td>
<td>181</td>
</tr>
<tr>
<td>setDebugFile</td>
<td>9</td>
</tr>
<tr>
<td>setDebugger</td>
<td>140</td>
</tr>
<tr>
<td>setDebugLevel</td>
<td>9</td>
</tr>
<tr>
<td>setDebugToFile</td>
<td>62</td>
</tr>
<tr>
<td>setDebugToFile</td>
<td>102</td>
</tr>
<tr>
<td>setDebugToScreen</td>
<td>62</td>
</tr>
<tr>
<td>setDebugToScreen</td>
<td>102</td>
</tr>
<tr>
<td>setEMS</td>
<td>40</td>
</tr>
<tr>
<td>setEMS</td>
<td>85</td>
</tr>
<tr>
<td>setEMS</td>
<td>134</td>
</tr>
<tr>
<td>setGtpAlarmReportingOff</td>
<td>293</td>
</tr>
<tr>
<td>setGtpAlarmReportingOn</td>
<td>293</td>
</tr>
<tr>
<td>setHeartBeatMenu</td>
<td>188</td>
</tr>
<tr>
<td>setIntendedRoute</td>
<td>310</td>
</tr>
<tr>
<td>setLastSelected</td>
<td>153</td>
</tr>
<tr>
<td>setLayerParameters</td>
<td>41</td>
</tr>
<tr>
<td>setLayerParameters</td>
<td>86</td>
</tr>
<tr>
<td>setManagedElement</td>
<td>42</td>
</tr>
<tr>
<td>setManagedElement</td>
<td>87</td>
</tr>
<tr>
<td>setManagedElement</td>
<td>143</td>
</tr>
<tr>
<td>setMessages</td>
<td>62</td>
</tr>
<tr>
<td>setMode</td>
<td>9</td>
</tr>
<tr>
<td>setModel</td>
<td>114</td>
</tr>
<tr>
<td>setModel</td>
<td>193</td>
</tr>
<tr>
<td>setModel</td>
<td>197</td>
</tr>
<tr>
<td>setModel</td>
<td>200</td>
</tr>
<tr>
<td>setModel</td>
<td>203</td>
</tr>
</tbody>
</table>
X

XkcdPanel ... 188
XkcdPanel ... 188
XmlParser ... 58
XmlParser ... 59
XmlParser ... 59
Appendix E – User Manual
EOS User Manual

TMF814 Simulator

Louisa Luciani, Mikael Riedel
Ericsson Lindholmen

E-mail Louisa: louisa.luciani@ericsson.com
E-mail Mikael: mikael.riedel@ericsson.com
Contents

1 Introduction ................................................................. 4
  1.1 Purpose ....................................................................... 4
  1.2 Intended Audience and Reading Suggestions ..................... 4
  1.3 Definitions and Abbreviations ...................................... 4

2 Overall Description ......................................................... 6
  2.1 Product Perspective .................................................. 6
  2.2 Operating Environment .............................................. 6
  2.3 Design and Implementation Constraints ........................... 6

3 Overview of the system ...................................................... 7
  3.1 System components .................................................. 7
  3.2 System dependencies ................................................ 8

4 Installation .................................................................. 9
  4.1 Database .................................................................. 9
  4.2 ORB .......................................................................... 12
  4.3 EOS .......................................................................... 13

5 Uninstall .................................................................. 14
  5.1 Database .................................................................. 14
  5.2 ORB .......................................................................... 14
  5.3 EOS .......................................................................... 14

6 Update .................................................................. 15
  6.1 Database .................................................................. 15
  6.2 ORB .......................................................................... 15
  6.3 EOS .......................................................................... 15

7 Launching .................................................................. 16

8 Adding objects ................................................................. 17
  8.1 EMS .......................................................................... 17
  8.2 MLSN ......................................................................... 17
  8.3 ME ............................................................................ 17
  8.4 PTP/FTP ..................................................................... 18
  8.5 CTP ........................................................................... 18
  8.6 TL ............................................................................. 18
  8.7 Alarm ......................................................................... 19
  8.8 TCA .......................................................................... 19

9 Removing objects ............................................................... 20
  9.1 EMS .......................................................................... 20
  9.2 MLSN ......................................................................... 20
  9.3 ME ............................................................................ 20
  9.4 PTP/FTP ..................................................................... 20
  9.5 CTP ........................................................................... 20
  9.6 TL ............................................................................. 21
  9.7 Alarm ......................................................................... 21
  9.8 TCA .......................................................................... 21

10 Updating objects ............................................................... 22
  10.1 EMS .......................................................................... 22
## Master thesis User Manual document for Ericsson OSS Simulator

### Ericsson Internal USER MANUAL

| 10.2 | MLSN                          | 22 |
| 10.3 | ME                           | 22 |
| 10.4 | PTP/FTP                      | 22 |
| 10.5 | CTP                          | 22 |
| 10.6 | TL                           | 22 |
| 10.7 | Alarm                        | 23 |
| 10.8 | TCA                          | 23 |

### NorthBound

11.1 Starting services                       24
11.2 Connect                                24

### Import data

12.1 Command-Line                           25
12.2 Through EOS GUI                        25

### Layout

13.1 Backup                                 28
13.2 Restore                                28

### Debug

14.1 Levels                                 27
14.2 Log to file                            27

### Database

15.1 Backup                                 28
15.2 Restore                                28
1 Introduction

1.1 Purpose

The purpose of this User Manual is to provide the user of the TMF814 Simulator with all the needed information to quickly and efficient setup and start using the simulator. It will also give a good overview of what the simulator are capable of doing. Under the corresponding subheadings, an explanation of the functionality will be provided, along with instructions and is some cases also pictures describing how to perform this functionality.

1.2 Intended Audience and Reading Suggestions

This document is mainly intended for Ericsson Solution Integrator, Solution Architect and Support Engineers to understand the functionality of the software. Begin with the overview sections and proceed through the sections that are most pertinent to you as a reader.

1.3 Definitions and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>TMF</td>
<td>TeleManagement Forum</td>
</tr>
<tr>
<td>MTNM</td>
<td>Multi Technology Network Management</td>
</tr>
<tr>
<td>TMF814</td>
<td>Protocol using CORBA specialized for MTNM communication</td>
</tr>
<tr>
<td>CTP</td>
<td>Contained Termination Point</td>
</tr>
<tr>
<td>PTP</td>
<td>Physical Termination Point</td>
</tr>
<tr>
<td>FTP</td>
<td>Floating Termination Point</td>
</tr>
<tr>
<td>TP</td>
<td>Termination Point</td>
</tr>
<tr>
<td>TL</td>
<td>Topological Link</td>
</tr>
<tr>
<td>MLSN</td>
<td>Multi Layer Sub Network</td>
</tr>
<tr>
<td>TCA</td>
<td>Threshold Crossing Alert</td>
</tr>
</tbody>
</table>
ME  Managed Element
EMS  Element Management System
2 Overall Description

2.1 Product Perspective

The GSDC OSS-integrations team works with customizations, integration services and solutions design of Network Management Systems. One of these systems is ServiceOn and often, a solution specific script will be created or an adjustment in the system will be made for the client. In order to verify the expected outcome, tests need to be performed on a physical network which is very expensive to have solely for testing purposes. This often means that tests are performed on-site at clients. The objective of this project is to create a standalone virtual network simulator that communicates northbound over TMF814. This product will simulate an optical network that can be interacted with through a GUI as well as through TMF814. The product is intended to be used for testing purposes, and will provide a way to test integrations in-house.

2.2 Operating Environment

The typical workstations the GSDC OSS Integration has are normal laptops. All the laptops have Operating system Windows Vista, this was the working environment and also the aim for simulator, although the simulator are platform independent. The laptop will typically have one or two gigabyte primary memory and a dual core processor with at least 1.5 GHz capacity.

2.3 Design and Implementation Constraints

The simulator and all the plug-ins are written in Java. This will run on all the computers used by GSDC Integrations and make it possible to continue developing the simulator by GSDC personal even after this project has ended.

An open source database will be used to handle the possible big amount of data. The choice has fallen on MySQL because it's free, works well with Java and makes the setup on each computer simple.
3 Overview of the system

The program is built according to the Model-View-Control architecture (MVC). The model contains the network data of the simulated OSS and encapsulates an internal database. The view displays all the information in the model, showing network elements, their correlation to each other and specific information about each network element. The controller receives input and makes calls to the model. Because the controller handles all the logic, which is preferably kept modular, it will consist of multiple classes. A main controller will for instance handle general logic such as the menu system, while specific windows are able to communicate with their own controller class. Multiple northbound interfaces are able to be loaded at startup, given that they follow the given NBI.

3.1 System components

- GUI
- Interface
- Database
- TMF814 implementation of NBI

![Figure 1, GUI main parts](image)
3.2 System dependencies

The program is divided into the components above, where the model is the central part. A graphical user interface that manipulates the model through views and controllers exists locally. The model can also interacted with through the northbound interface. The local GUI and the NBI should have the same functionality, though there might be some limitations in the NBI depending on the choice of protocol. If a new NBI is added to the simulator, functionality might have to be added to the GUI accordingly. The data in the database can be manipulated through either the GUI or the NBI.
4 Installation

To install and use this simulator three different parts must be installed on the system. The first two parts have to be installed before the last part.

4.1 Database

The simulator has support for MySQL database. MySQL version 5 or higher is required.

To install MySQL download the binary file, i.e. mysql-5.1.45-win32.msi.
For the latest version check http://www.mysql.com/downloads/mysql/

Follow the instructions in the installation-guide, remember your root-password, this is needed when you need to do any big changes to your MySQL-installation.

Choose to configure the database. If you want to register leave the registration box checked, otherwise uncheck this option.

When the database is configured correctly and is up and running open the MySQL Command-Line client from the start menu. Use your root password to login.

When that is finish run this command to create a user for the simulator:

-> GRANT ALL ON eosdb.* TO 'eos'@'localhost' IDENTIFIED BY 'pass';

Also run the command below to verify that the needed access has been assigned:
The important thing in the second result is that it should say GRANT ALL for eosdb.*. This means that the user is created in a correct way and the database is now ready to be used.

After point 4.2, when the simulator has been run with the flag -install, use these commands in the MySQL CL to verify the structure of the database:

-> use eosdb;

-> show tables;

The result should look like this:
<table>
<thead>
<tr>
<th>Tables_in_eosdb</th>
</tr>
</thead>
<tbody>
<tr>
<td>additionalinfo</td>
</tr>
</tbody>
</table>
| cct
| ctp                   |
| crossconnection       |
| ems                   |
| layerparameters       |
| layerrate             |
| managedelement        |
| nt_alarm              |
| nt_tca                |
| objectname            |
| ptp                   |
| severity              |
| supportedrate         |
| toplevelsubnet        |
| topologicallink       |
| x733additionalinfo    |
| x733monitoredattribute|

18 rows in set (0.07 sec)
4.2 ORB

All the packages needed to get a working ORB could be found inside OpenORB.zip.

Extract content of zip to i.e. c:\OpenORB\

It should now contain the following folders:
- EvaluatorUtility
- NamingService
- NotificationService
- OpenORB
- PersistentStateService
- tools
- TransactionService

These 7 packages could also be downloaded from OpenORBs official homepage:

http://openorb.sourceforge.net/downloads.html

Note that the latest version are hosted on another link, that could be found on the above page:

http://sourceforge.net/projects/openorb/files/

When the package are unpacked in i.e. c:\OpenORB\ it is time to set some environment variables.

To set an environment variable in Vista do this:

- Open start menu
- Right-click on computer, choose Properties.
- Choose Advanced System Settings, click continue on the UAC.
- Click on Environment Variables.
- Under System variables, click on new to create a new variable.

There are two variables needed, the first one is JAVA_HOME which should point to the folder where your java installation is. Inside this path you should see bin folder.

One example of how the path could look:

C:\Program Files\Java\jdk1.6.0_18
If you already have a path for the JAVA_HOME variable then you don’t need to do this.

The second variable that is required is the TCOO_HOME variable. This one should point to the OpenORB installation, that is to the folder that the content of the zip file is extracted, i.e. c:\OpenORB\.

Now all files needed by the ORB are installed. Only one thing remains, that is to tell your java-installation to use OpenORB as ORB for CORBA connections instead of the internal, not fully developed ORB. This is done by running a simple script.

- Open CMD.
- Move to the ORB installation:
  i.e. cd c:\OpenORB\
- Execute this command:
  java –jar .\OpenOrb\lib\openorb_orb-1.4.0.jar

This now creates a file named orb.properties inside the root of your java installation. If this file now exists the ORB is correctly installed and ready to use by any Java application.

4.3 EOS

Put the downloaded eos_X.X.jar in a folder on a path where you wish to have EOS installed.

Open cmd and move to the location of the jar file.

Run this command:
Java –jar eos_X.X.jar –install

X.X has to be changed to the version corresponding to the jar file.

Two files and a folder will be added to the path of the EOS jar file.

EOS is now ready to be used.
5 Uninstall

5.1 Database

MySQL has a built in uninstaller that can be used if the database should be uninstalled.

If problems occur and a removal of the database seems like the only option it is important to make sure that all files and all configurations are removed before the database is reinstalled.

5.2 ORB

To completely remove the OpenORB installation from your computer there are some steps that has to be done.

Remove the folder where all the files are.

Remove the environment variable TCOO_HOME.

Remove the orb.properties from JAVA_HOME

5.3 EOS

Since this simulator doesn’t really change any system setting the only thing needed is to remove the file that has been extracted. If everything is stored in one specific folder, i.e. c:\eos\ and no other program are installed there, then the whole folder could be removed to the trash-bin.
6 Update

Updating of the software is something that might have to be done from time to time. The part that will be updated most is EOS.

6.1 Database

If the change-log for EOS tells anything about updating the database program then this should be done by removing the installation and then installing the new one. Remember to grant access for the new user as in the installation part. Also run the simulator with the –install flag to create the needed structure inside the database.

6.2 ORB

The ORB hasn’t been changed very much the last couple of years, so a change here isn’t very likely. If something should be updated then just exchange the packages inside TCOO_HOME.

6.3 EOS

If it is only a minor update, then it’s enough to just replace the eos_X.X.jar file with the newer one. However if the changes are bigger, i.e. if something is changed on the northbound interface then the simulator has to be started with the install flag first:

Java –jar eos_X.X.jar –install

(X.X is the version of your downloaded file)
7 Launching

EOS could be launched in different ways. The simplest way after installation is to use the eos.bat file. This is just a batch file containing the needed information.

An alternative way is to start EOS through the command-line. EOS is started as a normal java program inside a jar-file. Change the directory in cmd to where the simulator is located with the cd command. Then use:

Java –jar eos.jar
for launching the program in normal mode

Java –jar eos.jar –install
to extract all the needed files and prepare the simulator to run in normal mode.

Java –jar eos.jar –parse
to enter parsing mode. This is a command-line interface to parse XML-files produced by the Script Client.
8 Adding objects

This section describes how to add different objects from the GUI.

8.1 EMS

EMS is the system that should be simulated, it could also be seen as an NMS, but under this simulator both goes under EMS. They contain the same information, and since the simulator operates over both management layers and have the functionality of both of them, they go under the same name.

To add an EMS to an empty system just right-click in the tree-window and choose new EMS. In the panel that pops up fill in at least name. The other fields are optional.

This is the only way to create an EMS from the GUI.

8.2 MLSN

MLSN are subnets for being able to manage a large amount of MEs. The can only be added in the left tree and having created the EMS first is a requirement.

Right-click somewhere in the white area or on the EMS to bring up the menu for EMS. Click on “Create MultiLayerSubnetwork”. In the popup only Subnet is required to be filled, this name has to be unique on the EMS. Other fields are optional. LayerRate requires a short, and will automatically fill in the name of the layer if it is predefined. The database contains 306 predefined names. Names of layers that are not predefined will only show as LR, over the protocol only the short is sent so the name is only for making it easier to see exactly which layer it is.

8.3 ME

Managed Elements are the lowest level in the tree to the left. They are also the nodes in the MAP-view. MEs could be created from the tree by right-clicking on the specific MLSN that is should be part of and choose “Create ManagedElement”.

MEs could also be created from the MAP-view by changing to editing mode in the map menu and then clicking somewhere on the graph where no node already are. If the node is created from the map, it has to be put into one of the available MLSN.
The name of the ME has to be unique on EMS level, which means that it’s not allowed to have two MEs with the same name but in different subnets.

### 8.4 PTP/FTP

The highest level of TerminationPoints. They could contain the exact same data and are only differed by the type ptp or ftp. They are created on a ME and at the moment this could only be done from the tree to the left.

A ptp/ftp is made unique by the EMS, ME and the type-name tuple.

The fields for Traffic Descriptor is not used by Ericsson at the moment, but the TMF814 protocol still allows this information.

If Transmission Parameters should be added it should be in the form of a short in the first field describing which layer rate, a string for name in the middle field and then an optional field for value. The combination of these values has to be unique.

When a ptp or ftp has been added it can be found under TP as main-view.

### 8.5 CTP

A ctp is a contained termination-point that has to have a parent TP. This could be another ctp or in the end a ftp or ptp. The combination of EMS, ME, parent and ctp-name makes the ctp unique.

Ctps are added in the TP main-view by right-clicking on an ftp or ptp and choosing "Create CTP". If the ctp is nested it should hold all the ctps in the ctp-name. ctp1/ctp2/ctp3.

The information stored for ctps doesn’t differ from ptpps or ftpps.

### 8.6 TL

Topological Links can only be added and viewed from the MAP-view. By choosing the editing mode in the menu and left-clicking on one node in the graph and dragging to another. If two nodes are successfully selected a popup window will appear with the information that could be stored on each TL. The name of the TL has to be unique on the EMS.
8.7 **Alarm**

Alarms can be added from all the objects that can through alarms just by right-clicking on them and choosing “Create Alarm”. There are not a name for each alarm that makes it unique but instead the combination of ObjectName, LayerRate, ProbableCause and ProbableCauseQualifier. No values are needed to be filled in, as long as the alarm is unique according to previous mentioned fields.

8.8 **TCA**

Resembles alarms in that way that they can be added from the same points. From the menus just choose “Create TCA” instead of alarm. A TCA is made unique by
9 Removing objects

By removing an object it will be removed from the database. This means that it can’t be undone. The simulator do not yet warn about this everywhere, so keep it in mind, and do not just click for the fun of it.

9.1 EMS

Removing the ems, that is removing all the information that is simulated could be done in different ways. The first way is from the menu -> File -> New Project. This will erase everything and give you the opportunity to start from scratch. The database could also be wiped clean, which means that all the information will disappear. This is done from the menu -> Database -> Clear.

None of them warns at the moment and all data will be lost.

9.2 MLSN

MLSN could be removed by right-clicking on them in the left tree.

Removing a subnet will remove everything that is underneath the subnet, which is MEs, TPs, TL connected to any ME in the subnet and of course all the information about the subnet itself. Removing a MLSN will however warn you about what the consequences are.

9.3 ME

ME could be removed by either right-clicking in the MAP-view on a node and choose “Delete Vertex” or just by right-clicking in the tree structure on the desired ME and choose “Delete ManagedElement”.

Trying to delete a ME will warn the user.

9.4 PTP/FTP

PTPs and FTPs are removed from the table in TP-view by right-clicking on the and choosing “Remove TP”. Warning will be displayed.

9.5 CTP

CTPs are removed just as PTPs and FTPs.
9.6 TL

Topological Links can only be removed from the MAP-view, this because this is the only view in the GUI where they are displayed. By right-clicking on the link and choosing “Delete Edge XX”. Warning will be displayed.

9.7 Alarm

Alarms can be deleted from the Real Time Alarm Monitor view. By right-clicking on the alarm or alarms and choosing “Delete selected alarms”. Many alarms can be deleted at ones. This is done without any warning.

9.8 TCA

TCAs are deleted in the same way as alarms.
10 Updating objects

All objects found by the GUI are easy to update. This means that to simulate two almost identical situations any of the value could easily be change for the second run and you don’t need to set up everything from scratch again.

10.1 EMS

Just right-click on the EMS or somewhere where no objects are in the tree view. In the menu choose to show “EMS info”. The same window for create the EMS is showed, but the bottom right button now says update. Change the values that you want to change and then click update. The information will be stored in the database directly.

10.2 MLSN

Right-click on the MLSN in the tree-view. Choose Subnetwork info. The rest is done exactly as in the EMS case.

10.3 ME

Right-click on the ME in the tree-view. Choose ManagedElement info. The rest is done exactly as in the EMS case.

10.4 PTP/FTP

Right-click on the PTP/FTP in the TP main view. Choose TP info. The rest is done exactly as in the EMS case.

10.5 CTP

Right-click on the CTP in the TP main view. Choose CTP info. The rest is done exactly as in the EMS case.

10.6 TL

Right-click on the link between two MEs in the map main view. Choose TopologicalLink info. The rest is done exactly as in the EMS case.
10.7 Alarm

Double-click on an alarm in the Real Time Alarm Monitor, or right-click and choose Alarm info. The rest is done in the same way as the EMS case.

10.8 TCA

This is done exactly as in the alarm case, but for the table of TCAs.
11 **NorthBound**

To be able to use TMF814 the NameService and the NotificationService has to be running. The services are normally started by running two scripts, but EOS gives the possibility to interact with the service from the GUI instead.

### 11.1 Starting services

To start the two services needed for the CORBA connection choose Services in the menu and click on start. After a few seconds the menu-dropdown will disappear and the status icon in the bottom right corner will turn green (this is the icon with "S" for Services). The service runs as a separate Java-thread and only one instance of EOS on each computer can start this service, otherwise you will get two services binding to the same port, port 21234 for NameService. This is not allowed by the Java Virtual Machine so the second EOS instance will be closed. Problems with connecting could have to do with that this port is already occupied. Make sure it isn’t if having problem.

### 11.2 Connect

To connect the actual northbound interface the services has to be running, otherwise a message will be displayed in the log-window depending on debug-settings.

One other requirement is that at least one implementation of the interface has to be found in the NBI directory. If no implementation could be found at startup, the NBI option in the menu will not be displayed. If one or more implementations are present they will be listed under the NBI menu.

To connect to a specific interface just hover the interface and select to connect. Another way to connect interfaces is to use the connect all option. This will connect all of the found implementation one after one.

When one or more implementations are connected, that is has a binding in the NameService and are ready to be used the status-icon in the bottom right corner will become green. The icon looks like an arrow pointing upwards for Northbound communication.
12 Import data

To be able to use real and accurate data it is possible to import data directly to the database through a parser that uses our database interface. This parser is just in beta version and some error-messages might not show enough information. The parser uses the information from the Script Client. The Script Client has some known problem to produce xml-files for some types, i.e. Alarms. Importing data will be done through a TMF-agent in the future, but as of now the parser is the only way to import much data.

12.1 Command-Line

The command-line parser can be used to import specific parts of the data, or all data depending on the choices the user does. To start the cl-parser just add a flag to the the launch of the eos.jar file:

```
Java –jar eos.jar –parse
```

This will not start EOS in normal mode but in the cl-parse mode. Just follow the instructions inside the parser.

When the parsing is done just run the simulator again, this time without the flag –parse.

12.2 Through EOS GUI

For convenience, it is also possible to import data from the Script Client from the GUI when running the simulator in normal mode. This is an easy to use extension of the parser that only imports as much as possible from a given directory (where the xml-files are). This feature requires that the database is empty before the parsing begins. Clearing the database could also be done from the EOS GUI.

The database features could be found under the menu Database.
13 Layout

EOS has to some extent the possibility to save layouts. By this we mean to save the appearance of the program. The layout could be restored to a default state. To save and restore the layout use the menu options for “Save configuration” and “Reset configuration” in the File menu.

The parts of the layout that will be stored by using the save option is which main-view should be the starting view (it takes the current), in which order should the columns appear in Alarm and TCA tables and what should the debug level be.

The settings are stored in the config.conf file that could be found in the root, where eos.jar is located.
14 Debug

Debugging is mainly for showing what is happening, a very good tool when something goes wrong. In the menu under Debug it is possible to set which level of debugging should be applied. The debug-messages are show in the status window in the bottom.

14.1 Levels

The levels follows the following pattern:

1 = All
2 = Finer
3 = Fine
4 = Config
5 = Info
6 = Severe
Off = Off

Depending on how much information the user wishes to see this level could be adjusted between these values.

14.2 Log to file

Logging to file is a feature to log all messages to a file called debugEOS.log. If another name for the file is wanted, this has to be changed in the config.conf file. The file will be found in the root, where eos.jar can be found.
15 Database

MySQL has some tools, that isn’t developed by this project and are not normally used by the simulator, but that could be very good to know about. I.e. it is very easy to backup and restore the entire database with one of the tools. By taking a backup of the entire database all the needed information about a scenario are stored to a file. This file could be restored on another computer or later in time on the same computer to restore the same network setup.

15.1 Backup

To backup the database a tool called MySQLDump can be used. This program makes a dump of the database, without any special flags it dumps both structure and contents of the database.

The syntax for using MySQLDump is:

```
mysqldump -u root -p -h localhost eosdb > eosdb.sql
```

This requires that MySQL bin-folder is part of the systempath, otherwise the absolute path has to be used to run MySQLDump.

```
C:\prog......SQL\bin\mysqldump -u root -p -h localhost eosdb > eosdb.sql
```

The result will be a single file, named eosdb.sql containing all the information in the database.

15.2 Restore

If you have a backup-file from an EOS-database it could be imported to restore to that exact state. This is done by simply running this command:

```
mysql -h localhost -u root -p eosdb < eosdb.sql
```

Note that this uses the ordinary MySQL and not MySQLDump. Good to know is also that this will erase everything in the database right now, so if something important is in the database just take a backup first, to another filename than the one that you want to load.