A MITEL
PRODUCT
GUIDE

Unify OpenScape Fault Management

Unify OpenScape Fault Management V12, OpenScape Desktop

User Guide
10/2021

A31003-F30C0-U113-01-7619

Mitel

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively “Trademarks”) appearing on Mitel’s Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively “Mitel), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively “Unify”) or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at iplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

Contents

Contents

1 Preface

This chapter discusses the following aspects:

- purpose and audience of this guide
- terminology
- organization of this guide
- conventions used in this manual

1.1 Purpose

This user guide describes the OpenScape Fault Management (OpenScape FM).

The OpenScape FM is a network and system management platform. This guide outlines the functions of the OpenScape FM and supports the user on his first steps within the management platform. The OpenScape FM is a modular software. It consists of a base module, which can be expanded with functions of the network and system management by initializing plugins.

This guide describes the graphical user interface and explains the usage of the OpenScape FM. The different available plugins are described in separate user guides. This guide addresses OpenScape FM beginners as well as advanced users.

This guide assumes that the user knows the basic concepts of TCP/IP networks and of network and system management.

1.2 Audience

This guide is addressed to end users who want to learn how to use the OpenScape FM Desktop.

1.3 Organization of this Guide

This guide is organized as follows:

- *Chapter 1, "Preface"* explains the structure of this manual.
- *Chapter 2, "Introduction"* provides information about the Desktop as network management application.
- *Chapter 3, "Basic Concepts"* provides some information about the theoretical concepts which form the basis for the OpenScape FM Desktop.
- *Chapter 4, "Getting started"* shows you how to start the Server and the Client.
- *Chapter 5, "The Client User Interface"* gives detailed explanations about all components of the Client UI.

- Chapter 6, “Server Configuration” explains the individual configuration options of the OpenScape FM Server.
- *Chapter 7, “Object and Event Search”* describes how to search objects in the OpenScape FM Desktop.
- *Chapter 8, “Event Actions”* explains how actions can be defined for individual events.
- *Chapter 9, “Display of Tray Bar Icons”* describes how to monitor an object in the Windows system tray bar.
- *Chapter 10, “Printing”* demonstrates the Desktop Printing functionality.
- *Chapter 11, “User Sessions”* explains the OpenScape FM Desktop User Session concept.
- *Chapter 12, “Creating Personal Views”* shows you how to build your own submap/object hierarchy as representation of the database objects.
- *Chapter 13, “Symbols and Status Display”* describes the Desktop symbols and the functionalities which are available via their context menus.
- *Chapter 14, “User and Group Administration”* provides detailed explanations for user and user group management.
- *Chapter 15, “Access Rights”* explains the access rights hierarchy and shows how to assign rights to users.
- *Chapter 16, “Network Topology Management”* gives information about the hierarchical structuring of your network representation and describes the connectivity of nodes.
- *Chapter 17, “Help”* explains the help facilities which are available via the Desktop.
- *Chapter 18, “Logging”* describes the OpenScape FM logging module for OpenScape FM applications and for OpenScape FM components.
- *Chapter 19, “Backup and Restore”* provides information about the backup and restore of the OpenScape FM database.
- *Chapter 20, “Time Schedule”* explains how to define and configure Time Schedules via the OpenScape FM Desktop.
- *Chapter 21, “Startup Manager”* describes the functions of the OpenScape FM Startup Manager.
- *Chapter 22, “Plugin Modules”* provides an overview about the different plugins.
- *Chapter 23, “Troubleshooting”* describes several possible failure causes
- *Chapter 24, “Database Files”* describes the OpenScape FM database files.
- *Chapter 25, “NAT Environment”* provides a description of how the OpenScape FM work in a NAT Environment.
- *Chapter 26, “HTTPS and Certificates”* describes how an HTTPS communication can be established between a OpenScape FM Server and Clients.
- *Chapter 27, “SSL Encryption”* shows the role of SSL in OpenScape FM
- *Chapter 28, “Mobile Access”* explains the OpenScape FM Mobile Access App for smart phone access.
- *Chapter 29, “String Formatting Language”* describes the language that can be used for the formatting of strings within various plugins.

1.4 Conventions Used in this Manual

The following font conventions are used in this document:

Bold Font: Indicates that a word is a new or important term. Bold also used for Buttons, menu names and item

Example: **Proxy Agent** or **OK**.

Bold Computer Font: Indicates data to be entered by the user.

Example: **`java`**.

Computer Font: Indicates computer output, including UNIX prompts, an explicit directory or a file name.

Example: `prompt%`.

Italics: Indicates a reference to another manual or to a different section within the current manual.

Example: *see IP Manager User Guide.*

Italic type is also used for emphasis.

Example: *All* users will be affected.

1.5 Terminology

- **OpenScape FM** means OpenScape Fault Management.
- **Server** means the OpenScape FM Server, i.e. the server where OpenScape FM Desktop has been installed.
- **Client** means the OpenScape FM Client, usually a web browser where OpenScape FM has been started.
- **Desktop** means the OpenScape FM Desktop.

Important Note:
We will use the term “Desktop” throughout the entire manual, but, please keep in mind that the Desktop comprises the OpenScape FM Framework AND the possibly initialized plugin modules. See also *Chapter 22, “Plugin Modules and Their Interaction”.*

2 Introduction

Companies have to rely on their IT infrastructure and on the business processes running within this structure. In such an environment it is important to identify problems with the infrastructure at an early state and to solve them before much harm is done. Or preferably, before they visibly effect the system.

The OpenScape FM is a system and network management platform that is able to monitor the IT infrastructure by using the IP Manager Plugin (see *IP Manager Plugin User Guide*). The System Management Plugin expands the IP Manager with functions that allow the monitoring of system parameters (see *System Management Plugin User Guide*).

With an adequate configuration of the IP Manager Plugin and the System Management Plugin it is possible to identify occurring problems at an early state and to solve them in a fast and systematic manner.

The OpenScape FM consists of a core module (the graphical user interface among others), into which the management plugins IP Manager and System Management are integrated. The core module provides a consistent access to the topology representation and the functions of both plugins.

A number of plugins is available that provide additional management functions and technologies like e.g. HiPath 4000 or OpenScape Voice.

2.1 Logical Architecture

Figure 1 provides an overview of the modular structure of the OpenScape FM. The core components offers basic features which are used by the other components (plugins).

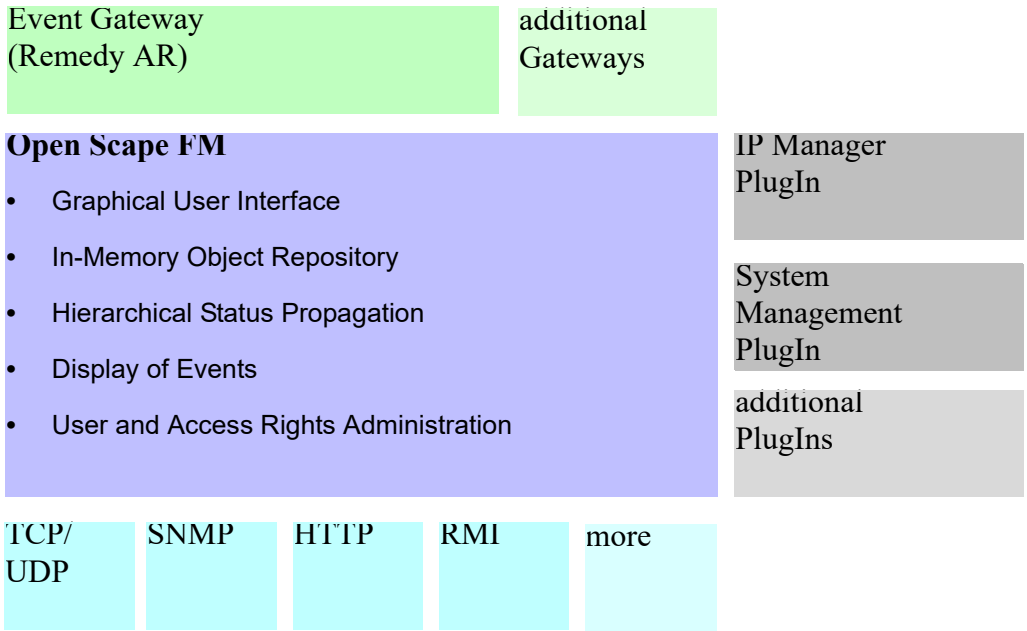


Figure 1 Logical Architecture

Through **Plugins** the OpenScape FM gets expanded with specific functionality. This will integrate specific management capabilities to monitored resources into the OpenScape FM.

Gateways allow the integration of other applications like e.g. Help Desk Applications or Workflow Processing.

2.2 Technical Architecture

The OpenScape FM consists of the management server and management clients which provide the GUI. This enables different users to work with the same management server simultaneously.

Figure 2 shows an example scenario, consisting of the OpenScape FM Server and two connected OpenScaoe FM Clients.

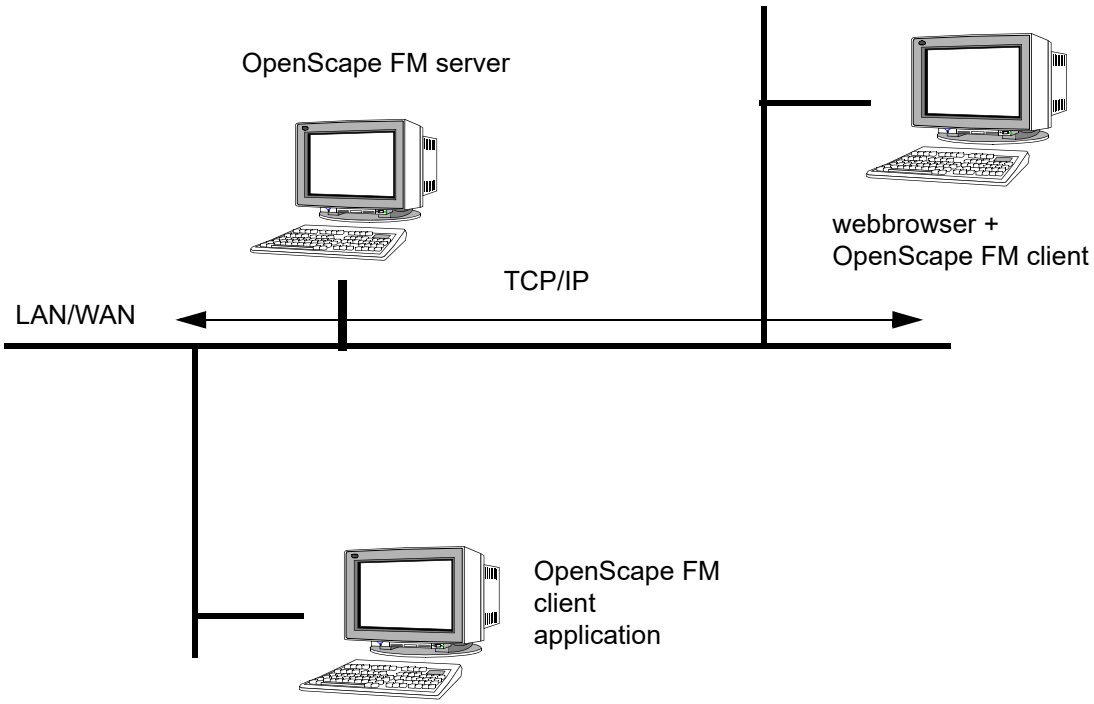


Figure 2 Client/server architecture

The Client is started as a Java applet in a web browser. However, the OpenScape FM also comes with a standalone client application that can be installed on a workstation.

2.3 Features

The OpenScape FM provides the following features:

- **Graphical User Interface:** The objects handled by the OpenScape FM Server will be displayed with the help of the OpenScape FM Clients.

- **Object repository:** All components managed by the OpenScape FM will be stored in the object repository. In the background, the in-memory structure will be stored in a persistent database, allowing a fast access by the Clients.
- **Hierarchically structured views:** Managed objects are structured in parent/child relations.
- **Easy navigation by means of submaps and object trees:** The managed objects and their topology are displayed on submaps and in tree views.
- **Status display by using colors:** A status is assigned to each managed object. The status can be compounded from the status of object's children.
- **Event Browser:** Events related to the managed components are displayed in a common event browser.
- **User and right administration:** To control the actions of users within the system, the OpenScape FM Server allows the administration of users and their rights.
- **Definition of specific views:** It is possible to define specific views. Selected objects can then be placed on these views.
- **Java-based Client:** The Client will be installed together with the Server. It will be running as a Java applet within a web browser. No further installation is needed to run the client.
- **Backup and Restore:** The OpenScape FM provides an automatic backup and restore mechanism. This allows the storage of the object repository and to load stored states into the system.

3 Basic Concepts

In this chapter, the basic concepts of the OpenScape FM Desktop will be discussed.

3.1 Database and Objects

The core component of OpenScape FM Manager is the database. All managed entities, like networks, subnetworks, servers, users, rights and others are stored as objects in this database, each with its specific object name. The entire database has a hierarchical structure, i.e. each object can have so called “child objects”. For example, given a server machine with several Ethernet adapters: each adapter would be represented by a child object of the server. Or a network “Germany” could have two subnetworks “North” and “South” which would be child objects of the network. Each object is stored in the database only once.

3.2 Symbols

Within the user interface, symbols are used to represent objects. Each symbol is defined by a shape, bitmap and label.

If a symbol type and label are defined for a specific kind of object, by default the symbol’s shape and bitmap indicate the object type to provide a quick overview over the various device types in the network. For example, a Windows NT PC would be represented by a rectangle with a Windows symbol at the center.

Symbols representing an element that is attached to another object/symbol are displayed on the submap of the parent symbol. For example, an interface symbol is placed below the symbol representing the host to which the interface is assigned.

Objects can be represented by more than one symbol, if these symbols are located on different submaps (see *Section 3.4.1*).

If an object is deleted, all symbols representing the object will be removed and the object is removed from the database.

If a symbol is deleted, only the individual representation of the object will be removed.

3.3 Events

Events describe occurrences in the OpenScape FM that are either brought to the system from outside (e.g., by SNMP traps) or that were detected by internal mechanisms of the OpenScape FM itself. (see *Chapter 8*).

Events often describe the occurrence or resolution of a problem. For example, an event can be triggered by the unreachability of an IP interface, and another event can describe that the same interface can be reached again.

Events are usually assigned to the object in the OpenScape FM that matches the content of the problem. In the example, this would be the object that represents the interface that was not reached.

For each event automatically a status that represents the severity of the problem gets assigned. For the non-reachability of the interface, this is the status '*Critical*', for the later determined re-reachability, this is the status '*Normal*'.

To each status a color is assigned. For ‚Critical‘ e.g. this is red, for ‚Normal‘ green (see *Section 5.11*).

Each new event is initially in the state 'Unacknowledged '. This state indicates that a problem is acute and should be displayed in the interface.

It can be set to the state ‚acknowledged‘ by a user or by automatic positive events.

The status and color of a symbol in the interface usually result from the 'worst' unacknowledged event assigned to the associated object.

An event can additionally be set to '*In progress*' by a user. This has no influence on the status of the object, but it is displayed to all users which person is currently taking care of the event.

Using the default server settings, confirming an event will automatically remove the attribute *In progress* from the event. This default setting can be changed within the server administration (**Server->Administration->Server Properties - Event Browser** - see *Section 6.4*)

3.4 Views

There are two ways of representing an object within its parent-child relations on the user interface: on a submap and in the tree view.

3.4.1 Submaps

A submap is a two dimensional graphical representation of all child objects of an object. Therefore an object’s submap can be opened to see if it contains child objects.

If e.g. the submap of a network object which contains three servers as child objects is opened, those three servers will be represented on the network’s submap.

Objects are represented by a symbol on a submap, but they can also be represented by different symbols on other submaps (see *Figure 3*). Changing a symbol‘s shape and bitmap, does not change the object it represents.

3.4.2 Trees

A tree is another form of representing the database objects within their parent-child relations. They are ordered in a tree structure which mirrors the object hierarchy. Each object is represented by the symbol which is characteristic for this position within the hierarchy. If you have changed a symbol at a certain location within the object hierarchy, it will be changed in the tree and submap representation at this location but nowhere else (*Figure 3*). Please read *Section 5.13, “Navigating Through the System with the Navigation Tree”* for a detailed explanation of the Navigation Tree.

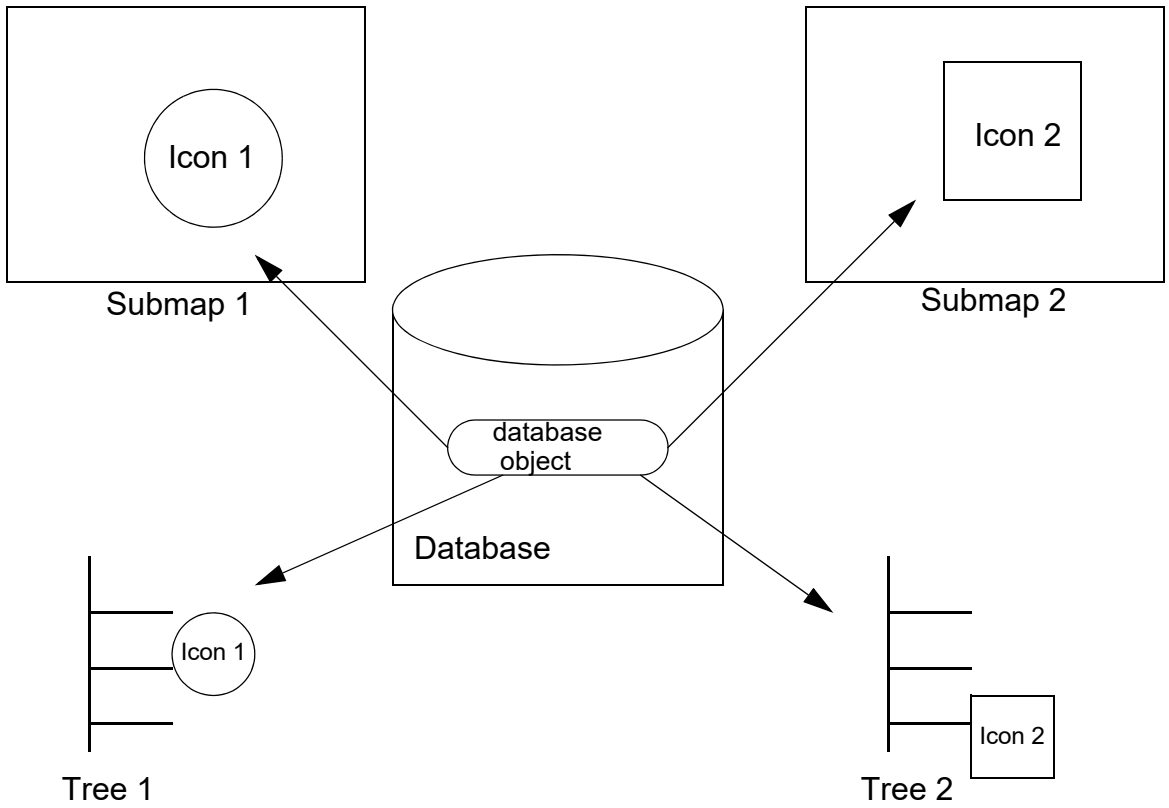


Figure 3 Representation of an object on two different submaps and trees

3.5 Maps

A map is a collection of submaps and trees representing the hierarchical structure of the managed objects. Many maps can exist at the same time. These different maps can be used to create different working environments which are characterized by the following criteria:

- Symbol position on the submaps
- Background image of submaps
- Scaling of submaps
- Icon types for the managed objects

Remember that there is only one database on a OpenScape FM server – all maps are based on the same stock.

3.6 Users

To enable access control, the OpenScape FM Desktop supports user and user group administration. A user is represented by an object in the database and has a login name and password assigned.A group is also represented by an object in the database. It is used to assign rights to a set of users at once. To learn the user administration, read *Chapter 14, “User and Group Administration”*.

3.7 Access Rights

The role of a user can be defined by assigning specific access rights. These access rights can be assigned either directly to a user or to a user group which the user is member of. Access rights can be assigned for a certain set of objects. It is possible to give a certain user access only to specific menu items or to assign a user administration rights for a specific object, but not allow access to other objects. For a detailed explanation about the administration of access rights, see *Chapter 15, “Access Rights”*.

4 Getting started

This chapter will discuss some basic aspects which should support the first steps within the OpenScape FM Desktop.

First the Server has to be installed. More about this can be found in *Appendix C, "Installation Process"*.

After this the Server can be started and one or more Clients can be used.

4.1 Starting the Server

OpenScape FM will be installed with the Startup Manager by default. The Startup Manager starts the OpenScape FM server after the installation automatically. More details about the Startup Manager can be found in *Chapter 21, "Startup Manager"*.

Appendix C, "Installation Process" gives a detailed description how to install OpenScape FM Desktop. A Java(TM) Runtime Environment (JRE) is required. A detailed description of the initial configuration can be found in *Appendix B, "Prerequisite Hardware and Software Environment"* and *Appendix D, "Configuration of the System"*.

4.1.1 Secure Data Transfer with OpenScape FM

By default OpenScape FM uses secure data transmission encrypting with the SSL/HTTPS standard. More about this can be found in *Chapter 26, "HTTPS and Certificates"* and *Chapter 27, "SSL Encryption"*.

4.2 Starting the Client

There are different methods to start the Client. These methods are described in the following paragraphs. Independant of the starting method, the same graphical user interface will be displayed.

4.2.1 Management via Web Browser

The standard way to access the OpenScape FM is via Java(TM)-capable web browser (*Appendix B, "Prerequisite Hardware and Software Environment"* for releases). In this case the URL of the OpenScape FM Server starts the landing page of the OpenScape FM.

`https://<OpenScape FM server>:3043/`

Selecting the button **Client Installer** installs a local OpenScape FM Client in the home directory of the current user and creates a Start Menu Entry and a Desktop Link. Both can be used to start the Java Client.
Clients installed with this method will be updated automatically.

The button **Portable Client** triggers a download that stores the Client as a file named `osfmPortableClient.exe` within the user's download directory. This file can then be copied to another computer, e.g. using a USB stick, to install the Client there.
Portable Clients will not be updated automatically.

The button **Web** starts the Web Client within a web browser.

The Java(TM)-plugin must be installed for the browser (refer to *Appendix B, "Prerequisite Hardware and Software Environment"* for the required version).

It brings up a browser window with the graphical user interface of the OpenScape FM Desktop. *Section 4.3, "Login"* describes the login procedure.

4.2.2 Management via Client Application

Alternatively the Client can be started as a standalone application.

The command **Start->Programs->OpenScape FM->Start Client (HTTPS)** starts the Client application using the Java(TM) runtime environment (JRE) from the path configured during the installation.

This activates the file `index.jnlp` which should be run with the **Java(TM) Web Start Launcher**.

If a Client is already running, a Client Application can be started by using the main menu entry **Client->New Client**.

4.2.3 Management via Unify Common Management Platform

Within the Unify environment, the OpenScape FM Client can also be started via the Common Management Platform (CMP).

For the initial setup of the Single Sign-on (SSO) within the CMP, an OpenScape FM user account with administrator rights is required.

If the connection to the OpenScape FM is initially set up within the CMP (**Maintenance->Inventory->Applications**) and saved, the OpenScape FM automatically accepts the certificate of the CMP web server on port `4709`.

The certificate can be viewed as usual in the OpenScape FM Client using the main menu entry **Server->Administration->SSL Certificates->Show Certificates** on the page **Trusted Certificates** (see *Section 27.3*).

4.3 Login

When the Client is started, a registration to the OpenScape FM Server must be performed. On the login page the OpenScape FM Server can be selected. In addition the RMI port for the connection can be entered. If no special port was configured for the Server, the default connection port to the OpenScape FM will be 3042.

4.3.1 Log in

To register to the selected OpenScape FM **Server** a user name (**Login**) and a matching **Password** are needed.

If the Caps Lock key is active while the password is being entered, a triangle appears to the right of the password entry field as a warning sign.

By using the selection menus **Language** and **Look & Feel** the appearance of the client can be selected.

Within the user interface of the Client the actual Server, the currently active user and the currently active map will be displayed on the upper right.

During the first login as user ’root’, i.e. after the installation process has been completed, a password for ’root’ must be defined (*Chapter 15, “Access Rights”*).

During the first login as a ’normal’ user, also a (new) password must be defined (*Chapter 14, “User and Group Administration”*).

The main menu entry **Client->Login** can be used to log-in as a different user. There is no need to restart the Client or to explicitly log out the current user.

4.3.2 Leaving the Client

To end the session for the active user, the main menu entry **Client->Logout** can be used.

The main menu entry **Client->Exit** will additionally close the Client.

5 The Client User Interface

This chapter describes the user interface of the OpenScape FM Desktop client. The first part provides some general information about how management data is displayed by the OpenScape FM Desktop Client and how to use common UI elements to navigate between different views of information.

In the remaining parts of this chapter, we take a closer look at four main components of the OpenScape FM Desktop user interface: The Topology Viewer, the Tree Viewer, the Submap Viewer and the Event Browser. You learn how to explore the network hierarchies and access information about managed resources.

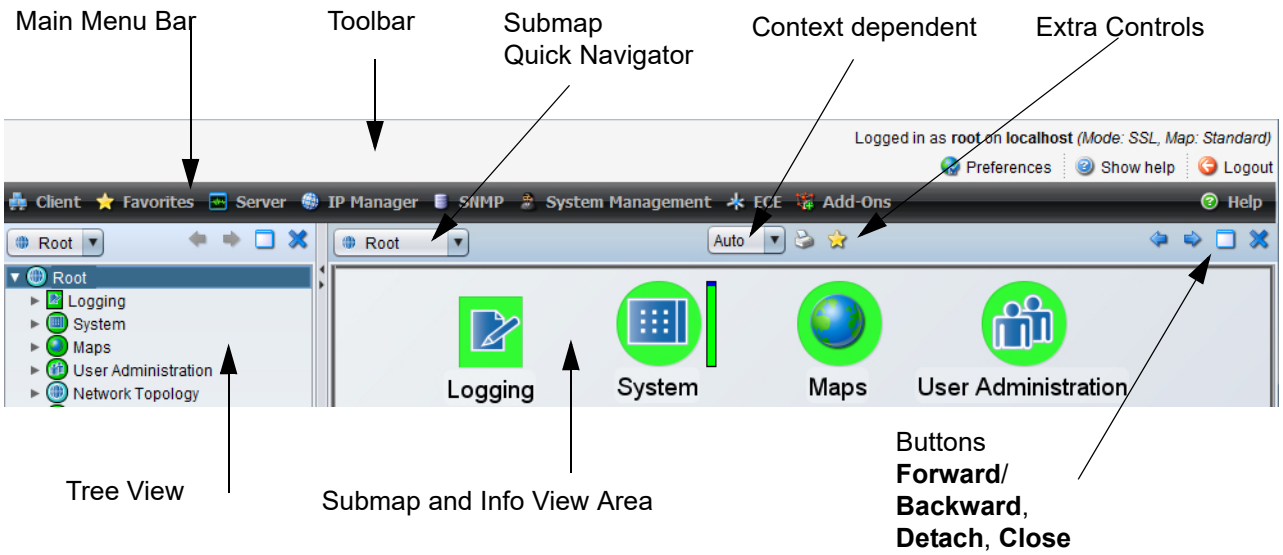


Figure 4 Structure of the Client User Interface

Figure 4 shows the window of the Client. It is divided into four parts. The top of the window contains the Toolbar (with name of the currently logged in user and the currently opened map on the left-hand side), below which you see the main menu bar. Below the menu bar, two areas where card panels are displayed are to be seen: the Tree View Area and the Submap and Info View Area. Both contain their Navigation Controls in the upper right corner: the **Forward** and **Backward** buttons, the **Detach** button, and the **Close** button. In the Tree View Area, one of one or several opened tree view(s) is displayed. The Submap and Info View Area displays different types of information, e.g. submaps, info browsers, the Message Log etc. When a new submap is opened, a new content is shown in the Submap and Info View Area, and this card is added to the opened cards of the Submap and Info Viewer. You can navigate back to the previous card by pressing the “Backward” button or you can switch to the next card by pressing the **Forward** button. It is also possible to detach cards from a card navigator with the **Detach** button – then this card is displayed in a separate window. The Quick Navigator of the Submap and Info View Area provides a list with all contained cards, i.e. all currently opened submaps, the Event Browser, and the Message Log.

In the following paragraphs, we will explain the menu components of the Main Menu bar and of the Submap and Info View Area in detail. The Topology Viewer uses the Submap and Info View Area in order to display network topologies – since this is one core component of OpenScape FM, we will explain the Topology Viewer in a distinct paragraph. Since the Tree View Area is a Card Viewer like the Submap and Info View Area, it has the same

Navigation Controls and functionalities as the latter, but because the Navigation Tree itself is a rather complex structure, it will be explained in detail in chapter *Section 5.13, “Navigating Through the System with the Navigation Tree”*.

5.1 Toolbar

On the right-hand side of the toolbar, the map that is currently opened and the name of the user who is currently logged in are displayed.

In addition the toolbar displays buttons to configure graphical settings, to display a help overview for all plugins and to logout from the client.

5.2 Main Menu Bar

The main menu is located below the Toolbar of the user interface. Depending on the initialized plugin modules, there may be additional menu entries. It has the following structure:

- **Client:**

  Root Navigation Tree: Opens a navigation tree beginning with the root object.

  Web: Opens the Web Client (see OpenScape FM Web User Guide) in a new browser window.

  New Client: This opens a new browser window displaying the login page of the Client.

  New Map: This command can be used to create a new map. A map name must be specified for this. If a new map has been created, a new object with the same name as the new map is added to the **Maps** container on the root submap.

  List Maps: This opens a list of all maps. From this list, a map to open or to delete can be selected. In a map, all views (i.e. submap and tree representation of the database objects) are grouped together, *Section 3.5, “Maps”*. In order to work with the Desktop Client, a map has to be opened first. The map which had been opened while logging out, will be opened by default, but another map can be opened via this menu item. Only one map can be open at the same time. If the Client is started for the first time and no map is created, a “Standard” map is created automatically.

  Object Search...: See *Chapter 7, “Object and Event Search”*.

  Preferences: This opens a window in which some client settings can be configured:

  - Within the panel **Looks** the general appearance of the client can be selected from a number of predefined **Look & Feels**.
  - Within the panel **Preferences** the first three possible selections define whether the respective elements (**Title**, **Login Details**, **Toolbar**) should be displayed within the upper part of the client window.
    If **Open Tree Path of Submap** is selected, objects whose submap is currently displayed are automatically selected within the Navigation Tree.

    The preferences **Close to system tray** and **Alarm sound for monitored object** configure the behavior of the monitoring by the Tray Bar icon (see *Chapter 9*).

Specifically, the behavior of the Tray Bar icon when closing the client is selected. Also, the warning tone played on deterioration of the status of the monitored object can be selected.

The **Auto Start Maximized** or **Auto Start Disabled** setting determines whether or not an OpenScape FM Client should be automatically started when a user logs on to the Windows interface.

Login: Use this menu option to login to the Server.

Logout: This menu item can be used to logout, i.e. to close your user session.

Change Password: This menu item allows the currently logged in user, to change the own login password.

Exit: Use this to logout from the Server and close the user interface.

- **Favorites:**

  Submap: If you have defined a home submap, this menu item will open it. (In order to define a home submap choose **Submap->Set Home** on a submap's background.) If no home submap has been defined, the Network Topology submap will be opened.

  Network Topology Navigation Tree: Opens a tree view starting with the Network Topology object. This tree contains all topology objects (see *Chapter 16*) and initially is the only displayed object tree.

  Network Topology Submap: Opens the submap of the Network Topology object. All other topology submaps can be accessed from this submap.

  Events: Via this menu item the Event Browser can be opened. Another method to do this is to use the Quick Navigator of the Submap and Info View Area.

- **Server:**

  Administration:

  - **Server Properties:** This opens a window with tabs to define the email default parameters, define the connection method for clients, configure the Active Directory and display some general information about the server. More about this can be found in *Chapter 6*.
  - **User Administration:** This displays the User Administration tree in the tree window. More about user administration can be found in *Chapter 14, "User and Group Administration"*.
  - **Debug Options...:** here a user with Administrator rights can activate the debug and log mechanism for the plugins, see *Section 23.2, "Log and Debug Configuration"* for more details.
  - **Domains:** here, you can open a list with all defined Domains, see *Section 15.8.4, "Overview Over All Domains"*.
  - **Backup Manager:** see *Chapter 19, "Backup and Restore"*.
  - **Startup Manager->List Services:** here, a user with "Administrator" rights can view all services which were started by the Startup Manager and also can stop, start or restart such a service, see *Chapter 21, "Startup Manager"*.
  - **License Manager:** see *Appendix E, "Licensing"*.

  Time Scheduler...: here time schedules can be defined and reconfigured. More about time schedules can be found in *Chapter 20, "Time Schedule"*.

Plugins: here, a user with “Administrator“ rights can initialize the available plugins, see *Appendix D, “Configuration of the System”*. When all plugins are initialized the menu item is not available any longer.

- **Help:**

  About: This opens the about window where further information about the installed OpenScape FM and copyright and trademark information are displayed. The about window can also be opened via the Submap Quick Navigator (*Figure 4*).

  Basics: This opens the online help for the OpenScape FM Desktop. If there are other active OpenScape FM plugins, the respective entries are to be found and open the corresponding online manuals. *Chapter 17, “Help”*.

5.3 Main Menu Objects

On the root submap you will find two objects, which represent specific entries of the main menu bar: **Maps** and **System**. These objects in combination with the use of access rights enable an administrator to set up user accounts with very specific constraints. So a user with Administrator rights can create another user who has for example only one menu item in the IP Manager main menu. For more details about access rights please refer to *Chapter 15, “Access Rights”*.

5.3.1 Maps

The Maps icon contains all existing Maps (for more details about maps please read *Section 3.5, “Maps”*). The Maps icon offers next to the common object menu items the following menu items:

- **List Maps** is the same like the menu item **Client->List Maps...** in the main menu bar. It lists all existing maps and you can open and delete maps within this browser.
- **New Map** is the same like the menu item **Client->New Map...** in the main menu bar. It opens a window where you can create a new map by specifying a map name.

By default the Maps icon contains at least the **Standard** map, which will be created by the system. When you create another map a new icon representing this map will be placed below the **Maps** icon. The icons on the submap of the Maps icon offer in addition to the common object menus in their context menu the following menu items:

- **Open Map** opens this map, it works exactly like the **Open** button in the **List Maps** browser.
- **Delete Map** deletes this map, it works exactly like the **Delete** button in the **List Maps** browser.

5.3.2 System

The **System** submap contains the icons **Server**, **Plugins** and **Help**.

- The **Server** icon offers, additional to the common object menus (*Section 5.6.1, “Object Context Menus”*), the same menu items as the **Server** menu in the main menu bar. For description of the functions please refer to the *Section 5.1, “Toolbar”*. Below the Server icon you find objects for functions which are supported by the OpenScape FM Desktop. These objects are the **Backup Manager**, the **Startup Manager** and the **License Manager**. According to the function/object you may find also objects below that one. All the objects below the Server objects propagate their menu items up to the Server main menu. For more information about these functions please refer to the offered chapter in this guide:
License Manager please see *Appendix E, “Licensing”*.
Backup Manager please see *Section , “Backup and Restore”*.
Startup Manager please see *Chapter 21, “Startup Manager”*.
- The **Plugins** icon offers only the common object menus. Within the **Plugins** submap you will find objects which represent the initialized plugins. The context menu of each plugin object has the same menu items as the dedicated plugin menu entry in the main menu bar.
- The **Help** icon contains a child object for each available help category and an "Info" icon for "About". In the context menu of the Help icon an entry allowing to open the corresponding help and additionally an menu item for opening the "About" window are available. The respective child objects of **Help** representing an help category are named like the initialized plugin they belong to and open (over the menu item **Help**) only the help of the dedicated plugin. The "Info" icon offers the menu item **About** and opens the "About" window.

5.4 Submap and Info View Area

The Submap and Info View Area is always visible. It has six elements which are described in this section: the Quick Navigator, the Extra Controls “Zoom Selector” and “Print” Button, the “Forward”/”Backward” buttons, the “Detach” button, and the “Close” Button (see *Figure 4*).

All elements of the Navigation Tree (in the Tree View Area) will be covered in a separate section, *Section 5.13, “Navigating Through the System with the Navigation Tree”*.

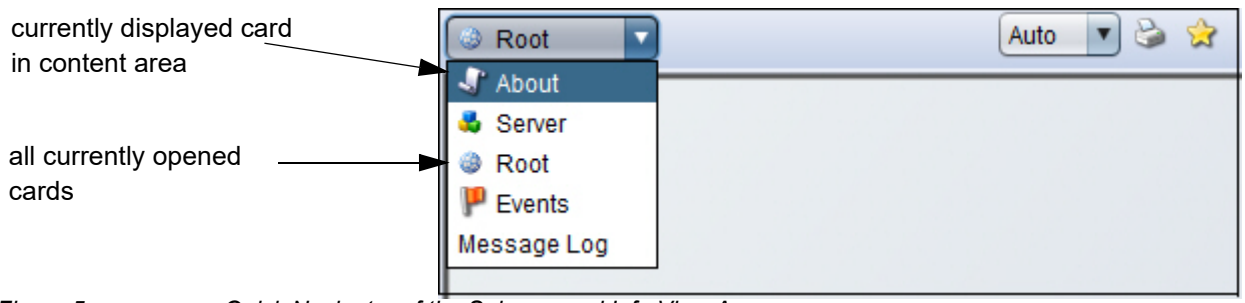


Figure 5 *Quick Navigator of the Submap and Info View Area*

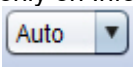
Quick Navigator (*Figure 5*):
The OpenScape FM Desktop server also maintains a list of all cards (i.e. submaps and other info panels) currently opened. Detached cards are listed in brackets. You can access the submap-list using the quick navigator, located at the upper left. A selection of a submap from this choice will open it immediately. When you select a detached card, it will be put to the front on the screen. When a submap is closed, it disappears from the list in the Quick Navigator. However, some cards cannot be closed, like the Message Log and the Event Browser – thus they will always be available via the Quick Navigator’s pull-down menu.

Button Forward/Backward: 
The Backward button opens the previous card in the list of opened cards. The Forward button switches back to the submap that was displayed before you used the backward button.

Button Detach: 
By default, the Client displays all cards in the Submap and Info View Area of the client's main window. If you want to keep a submap or an Info Browser open in a separate window, you can use the "Detach" button which opens a new window containing the current card. You can even assign a card title and/or symbol for the minimized card window, see *Section 5.5, "Submap Icons and Submap Titles"*.

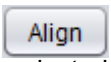
Button Close: 
Use the close button to close the current submap or data browser. It will be closed in the content area and will not be offered via Quick Navigator anymore.

The following two components are located in the context dependent area for Extra Controls. They only appear on submaps (Zoom Selector and "Print" Button) or only on Info Browsers!

Zoom Selector (only available on submaps): 

With the zoom selector you control the size of the symbols/networks displayed by the topology viewer. You can select different zoom factors by steps of 25 percent between 25 and 200 percent. If you select "Auto", the display of the network is always matched to the window size.
If "Auto-scale" is switched off, you can resize a submap manually by dragging its borders with the left or right mouse button. Left: all symbols will keep their positions relative to background. Right: all symbols will keep their real position.

Switch Align To Grid (only available on submaps) 
If this switch is activated, symbols can only be positioned on grid points when manually moved. If a symbol is moved manually while the switch is activated, the symbol automatically moves to the nearest grid point after positioning.

Button Align (only available on submaps) 

This button can be used to align the currently selected symbols of the submap with each other. When the button is pressed, a menu opens with the various alignment options, whereby the alignment is always based on the enclosing rectangle around all selected symbols.
The symbols can be moved to one of the edges of the rectangle, or they can be distributed evenly along the X or Y axis.

Button Screenshot 
This button copies the display area of the currently displayed submap to the clipboard.

Button Reload 
This button updates the contents of the currently displayed submap. For example, the data for a shown table is refreshed.

Button Print:(only available on submaps and in Info Browsers): 
On some cards the "Print" button is to be found between the Quick Navigator and the "Backward"/"Forward" buttons. Use this button to print out a copy of the display area of the currently displayed submap. See *Section 10.1*.

Button Select All: (only available in Info Browsers): 
With this Button you can select all rows listed in the opened Info Browser and proceed an action for all these rows, alternatively you can also press Ctrl.+A. Some browsers does not support multiple row selection, in these browsers the **"Select All" Button** will not be availab**le.**

Button Copy: (only available in Info Browsers): 
This Button copies the content of the selected rows into your clipboard, alternatively you can also press Ctrl.+C. To copy also the heading line of the info browser hold the shift key down while you press the copy button or press Ctrl.+Shift+C. To copy all rows without selecting use Alt+C. The string values of the selected rows will be separated by a tabulator character and the rows by the system specific line separator.

Button Help: (only available in Info Browsers): 
This button opens the context sensitive help to the opened info browser, see also *Chapter 17, "Help"*.

Button Favorites: 
This button adds the current view to the list of favorites (see *Section 5.15, "Favorites"*). Depending on the configuration of the user's Start View (see *Section 14.1.1*), the favorites are assigned to the user or his Start View group.

5.5 Submap Icons and Submap Titles

Cards managed by the Quick Navigator can be detached and displayed in a separate window (see *Section 5.4*, **Button Detach**).

In general, the title of a detached card is its standard symbol label. If the detached card is minimized, by default the OpenScape FM logo is displayed as its icon.

For cards that represent submaps and trees, an individual card title and an individual card symbol can be defined, which are displayed when the respective card is detached. The card title is assigned to the parent object of a submap or the root object of a navigation tree.

Symbol and card title are defined using the user interface for the card display. To define a symbol and/or a card title for an object, the entry **Properties** can be selected from the context menu of the object and the page **Symbol->Card Display** can be opened.

The page contains the user interface for the card display.

On the left side of this window is a list of all symbols that have already been loaded. If one of these symbols should be used for the current object, it must be selected and **OK** has to be clicked.

If another icon should be added to the list, this can be done with the help of the button **Upload Icon**. In this case a file browser opens in which the desired icon can be searched and selected. Currently the formats gif, jpg and xpm are supported. The selected image file is inserted into the list of card symbols and can be selected and assigned there as described above.

Clicking the **Clear** button will delete the configuration. An icon file can be removed from the OpenScape FM by selecting its name in the list and then clicking the **Delete Icon** button.

In the right part of the window, the **Card Title** for the detached card can be set.

In addition, it can be specified here whether the configuration should apply to the current map only (**Map-Specific**) or to **All Maps**.

The selected icon for a submap and the card heading are displayed when the corresponding submap is detached (right/top button) and then minimized.

5.6 Context Menus

5.6.1 Object Context Menus

Object context menus are activated by pressing the right mouse button on symbols on a submap or in a tree. The context menus are context sensitive. That means that the contents of a context menu differs, depending on the object from which it is activated. All symbol and submap contexts have some options in common and add some symbol-specific options.
Furthermore, the access-rights of the user and the context influence the appearance of a context menu. If the user does not have the right to use a certain functionality, the corresponding menu items are not displayed.
In this chapter we explain the general available object menu items:

- **Events:**

  Acknowledge...: acknowledges all events of this object if events for this object exist. When the object has a compound status also the events of its child objects will be acknowledged, see also *Section 5.9, “The Event Browser”*. For more details about the compound status please refer to *Chapter 13, “Symbols and Status Display”.*
  Hint:
  If an event is acknowledged that is also assigned to other objects, the confirmation also affects these objects.

  View...: opens the Event Browser with the object-specific entries. This menu item is only available when events for this object exist.
  When the object has a compound status this menu item will be shown although this object has no events. In that case, by selecting this menu item you will get all the object-specific event entries of the child objects, too. For more details about the compound status please refer to *Chapter 13, “Symbols and Status Display”*.
  To see again all events in the Event Browser you should choose the event category “All Events“.

- **Open:**

  Object Search...: opens the Object Search Dialog where you can search after a certain object below the selected object. Please see *Chapter 7, “Object and Event Search”* for more information.

  Tree: sets the selected object as root of a new tree and displays it in the Tree View Area.

  Submap: opens the object’s submap. You can either use this option or double-click on the object.

  Tree Path: opens the tree view and selects the current object in the tree (not available within the Tree View Area).

- **Edit:**

  Hide: sets the symbol to hidden (see *Section 12.1, “Hidden Objects”*).

  Show Hidden Objects: sets all hidden symbols on the submap to visible (see *Section 12.1, “Hidden Objects”*).

  Copy: prepares the object for a copy, see *Chapter 12, “Creating Personal Views”*.

  Cut: prepares the object for a move.

Paste: If an object was prepared for a copy, this object will be copied to the submap of the current object if possible. If an object was prepared for a move, this object will be moved to the submap of the current object if possible. Please keep in mind you can not add an object manually on application created submaps, see also *Chapter 12, "Creating Personal Views"*.

Link: creates a connection between two symbols, see *Section 5.11.1, "User Defined Connections"*.

Set Target: sets the target of the created connection [is only usable when a connection is created], see *Section 5.11.1, "User Defined Connections"*.

Clear Connection: clears the created connection, [it is only usable when a connection is created], see *Section 5.11.1, "User Defined Connections"*.

Delete Object...: deletes this object and all symbols that represent this object.

Remove Symbol: removes the individual symbol.

Unmanage/Manage: unmanages/manages this object. If an object will be set to unmanage, technology type specific menu items will be deactivated. Furthermore IP polls will no longer be performed for this object, i.e. no status and configuration changes will be shown for this object. Events will be ignored. In some cases also child objects which offer additional functionalities will be deleted. As soon as the object will be set to manage the technology type specific menu items and objects will be accessible again and the IP polls for this object will be activated. If the manage/unmanage operation will be done on IP nodes the manage/unmanage operation will be proceeded also for the child objects of this IP node.

- **New:** This menu contains entries for the various object types that can be added to the submap of the current object. If a menu item is selected, an object of the respective type will be added manually to the submap.
- **Configure:** This will open a window with means to configure the current object. The content of the window depends on the object type and the discovered object properties.
- **Properties:** Opens a window containing, among others, the following pages:

  Domains: here an object can be assigned to a Tenant Domain. *Section 15.8, "Domains"* provides further information about this.

  Symbol->Properties: opens the Edit-Properties GUI where the symbol's view and map-specific label can be modified, e.g. a new shape can be associated. See also *Chapter 13, "Symbols and Status Display"*.

  Symbol->Card Display: opens the Card Display GUI where an icon and/or a title for an object can be chosen, see *Section 5.5, "Submap Icons and Submap Titles"*.

  Comments: as an "Administrator" or an "Operator", can add a comment to this object here (see *Section 5.16, "Object Annotation"*).

  Info: opens the "Info Browser" which displays general object properties. Please see *Section 5.7, "Standard Info Browsers"* for details about Browser functionalities.
- **Status Explanation:** Explains the current status of the object (see *Section 5.17, "Status Explanation"*).

5.6.2 Submap Context Menu

The context menu of a submap corresponds to the submap of the object to which the submap belongs. The various functions behave as if they are performed for the respective object.

An exception are the functions **Delete Object** and **Remove Symbol**. These cannot be performed from the submap‘s context menu, but only from the context menu of the object itself.

In addition a number of submap specific functions exist. These are:

- **Up:** opens the parent submap from which this submap has been opened.
- **Down:** opens the child submap from which this submap has been opened.
- **Submap:**

  Refresh Layout: aligns the symbols on the submap in a way that they do not overlap.

  Layout: In this menu it can be selected how connected objects should be positioned automatically. For unconnected objects, these selection has no effect.

  Set Home: sets this submap as Home submap. The Favorite **Submap** can be used to navigate directly to this submap.

  Properties: opens the Submap Properties GUI to select a default background image for all submaps and a background image for the actual submap. See *Section 5.8, “Selecting a Background Image”* for more details.

  Set Monitor Object: sets the monitor object for the Tray Bar Icon of the current user. For more information about the Tray Bar Icon please refer to *Chapter 9, “Display of Tray Bar Icons”*.

5.7 Standard Info Browsers

When you open an object’s context menu and select **Properties**, a browser which lists basic information about this object on the page **Info** is opened. But this is only one of several places where you will encounter those Info Browsers. They offer some functionalities which make the work with OpenScape FM Desktop and its plugins quite comfortable:

You can sort the browser’s contents by clicking on the corresponding table heading. To set the sorting back to default, click on the corresponding table heading while pressing the "Ctrl" key.

If the browser has more than one column, you can put the table/browser columns in any order you like by dragging the heading (and with it the entire column) to the desired position with the mouse. OpenScape FM will store this configuration as user-specific data, thus when you have arranged the columns in a certain order, the next time when you open that browser, you will find all table entries at exactly the place where you put them.

If a column is to narrow to show the complete information, you can click the corresponding header while pressing the "Shift" key: the column will then be set to the ideal column width so that the complete column header and all information are visible.

In order to set the ideal column width for all columns of the info browser at once, you have to mark one row and press the keys "Ctrl" and "d" at the same time.

Plugins like the HiPath 4000 or the HiPath 3000/5000 use a great number of Info Browsers in order to represent up-to-date information about the managed components. Often you will find a **Stop** button to stop the current request, and a **Reload** button to start a new request and display the up-to-date data.

Above the list, the button **Copy** can be used to copy the content of the selected rows into the clipboard.

The button **Print/Save table** next to it opens a window which can be used to export the data displayed in the Info Browser. **Print** sends the data to a printer. **Save** will export the data into a file using a comma separated values (csv format).

The following table contains a list of the available key commands. The input focus must be set to a browser line to activate the functions.

Defined Input	Function
F1	Starts the Online Help
F3	Search forwards (after a search string has been defined)
Shift + F3	Search backwards (after a search string has been defined)
F4 oder ALT + F4	Close the Browser
F5	Refresh the content
CTRL + D	Optimize column widths
CTRL + E	Delete all
CRTL + F	Define a search
CTRL + L	Set layout to max width
CTRL + klicken	Go back to the default line order
X	Cut
Y	Copy

Table 1 Keyboard functions within Info Browser

5.8 Selecting a Background Image

For each submap in the OpenScape FM environment a different submap background image can be configured. This image will be displayed as a background whenever the submap is opened.

In addition a single default background image can be selected for a map. This default background image will be displayed for all submaps of this map, for which no specific submap background image has been configured.

Important Note:
The assignments of specific or default background images are only valid for the active map. It is even possible to define different background images for the same submap in different maps. (see *Section 3.5*)

The background images can be used to add further information to a submap. E.g. a country map or the layout of a factory can be used. Objects can then be placed on the submap in relation to their position in real life. Or simply the picture of e.g. a nice sunset can be used to lighten up the look of the management platform.

Both, the configuration of submap and the configuration of default submap background images, is done in the "Submap Background Manager" window, which will be opened by selecting the **Submap->Background** menu item from the context menu of a submap.

The "Submap Background Manager" window consists of the following elements providing the following functionalities:

- The **Background Image** list is the list of images that have been uploaded by a user. Only these images can be used as background pictures. When the "Submap Background Manager" is initially opened, the defined background image of the active submap will be selected in the list.
- The **Upload Image...** button can be activated to add a picture to the Background Image list. When this button is pressed, a file selector window opens in which a picture can be selected. When the file selector window is closed by pressing its "Ok" button, the currently selected picture will be added to the list. Currently GIF and JPEG picture formats are supported. Selecting another picture format or selecting a non picture file will result in an error message.
- The **Scope** selector can be used to define the scope for the next "Ok" or "Clear" action as described below. When the "Submap Background Manager" window is opened, the scope is set to "Submap".
  - When the scope is set to **Submap**, the image currently assigned to the active submap will be selected in the Background Image list. The next "Ok" or "Clear" action will only apply to the active submap.
  - When the scope is set to **Global**, the image currently defined as the default background image will be selected in the Background Image list. The next "Ok" or "Clear" action will apply only to the default background image.
- When the **Save** button is pressed, depending on the selected scope, the following will happen:
  - When the scope is set to **Submap**, the image currently selected in the Background Image list will be set as the specific background image of the currently active submap. The selected image will be displayed.
  - When the scope is set to **Global**, the image currently selected in the Background Image list will be set as the new default background image. It will be displayed on all submaps for which no specific background image has been assigned.

  In both cases the "Submap Background Manager" window will be closed.
- When the **Clear** button is pressed, depending on the selected scope, the following will happen:
  - If the scope is set to **Submap**, no image will be assigned as the specific background image of the currently active submap. When a default background image has been defined, it will be displayed instead. If no default background image exists, no background picture will be displayed for the active submap.
  - If the scope is set to **Global**, no picture will be assigned as the default background image. Submaps which have no specific background image will no longer display a background image at all.

  In both cases the "Submap Background Manager" window will be closed.
- The **Delete** button when pressed will remove the picture currently selected in the Background Image list from this list. The removed picture will no longer be displayed as a background or the default background image. Besides the removal from the list, deleting a picture shows the same results as pressing the "Clear" button for the picture in both scopes.
- The **Cancel** button will close the "Submap Background Manager" window and no action will be executed. Already uploaded or deleted pictures will stay uploaded or deleted.

5.9 The Event Browser

The Event Browser is the central tool to display all events within the OpenScape FM.

When the OpenScape FM is opened, the Event Browser will be displayed in the view area, displaying all events the current user is allowed to see. The main menu entry **Favorites->Events** can be used to display the Event Browser, displaying all events, at a later time.

The major part of the Event Browser consists of a table in which each row represents an event. A colored bubble in the column **Severity** within a respective row represents the severity assigned to the event.

A double click on a row that represents an event to which an object is assigned, opens the submap on which that object is located. If no object is assigned, a view opens which provides information about the current event. This information can alternatively be displayed for all events by selecting **Event->Event Data** from the context menu of an event.

Note:
Older versions of the OpenScape FM represented the Severity with completely colored rows. This old behavior can be globally reactivated by entering the line
`server.colored.events=true`
within the file
`<OSFM-install-dir>/startup/conf/OpenScapeFM.properties`

The different columns display the information, whether the event has been **Acknowledged**, whether it has been set to **In Progress**, whether an **Annotation** for the event exists, the **Severity** assigned to the event (displayed in the respective color), the **Date** when the event was received, the **Source** and **Category** of the event and a short **Description** of the event. The last two columns display the number of already existing events correlated to the event (**Correlated Events**) and the number of events of the same type (**Type Count**).

By clicking on the column headings, the contents of the table can be sorted in ascending or descending order according to the clicked column. This sort order is saved for the user and will be used automatically the next time he logs in.

If an **Annotation** has been assigned to an event, the respective column displays a speech bubble. The tooltip for this column then displays the content of the annotation and the tooltip for the column **Acknowledged** displays the last annotation modification time.

Events are correlated, if they are registered as child events. These events use the same internal event key for the same object (e.g. different state events for the same object). The events that are correlated to an event can be displayed by using the menu entry **Event->Correlated Events** from the context menu of the event.

Whether correlated events exist for a specific event and whether it is acknowledged can be identified by the symbol that is shown in the **Acknowledged** column. For individual alarms, a **single green check mark** (for acknowledged events) or a **single red square** (for unacknowledged events) is displayed. If correlated events exist, the symbols are accordingly changed to **two green check marks** or **two red squares**.

Events are of the same type, if they triggered the same reaction for the same object (e.g. two Major Alarm On events for the same object). Events that have the same type as an event can be displayed by using the menu entry **Events->Events of Same Type** from the context menu of the event.

The buttons below the table can be used, to **Acknowledge**, **Unacknowledge**, set to **In Progress**, set to **Not in Progress**, **Annotate** or **Delete** selected events, if the current user has the necessary rights.

Since there will be a large number of events, filters can be used to reduce the number of displayed events to a wanted subset.

The pages located above the table can be used, to define a search within the displayed events. Each entry will reduce the displayed events to only those that match **all** entries on **all** pages. By default, four pages are always available, more pages can be added by plugins for plugin specific search criteria.

The page **Properties** allows the search for the **Acknowledged** status, the **In Progress** attribute, the event **Category** (the menu displays all categories for which events are available), the **Description** and the **Source** of the event. Entries in the last two fields will reduce the search to substring matches in the respective columns.

An entry on the page **Fulltext** will perform a substring search in the columns **Description**, **Category** and **Source**. One of these has to match for the displayed events.

The page **Severity** can be used to limit the search to the checked severities.

And the page **Time/Maintenance** can be used to limit the search to events which are received within a given time set. A minimum date (**Start Date**), a maximum date (**End Date**), a **Time Schedule** and a **Maintenance Status** (Maintenance Filter) can be selected.

To avoid the separate listing of correlated events, the checkbox **Correlated** on the page **Properties** is set by default. Only the last of the correlated events will then be displayed. The checkbox can be unchecked to see all correlated events, but the default itself cannot be changed. If another Event Browser is opened, the check box will be checked again.

To see the correlated events to a single event, the menu entry **Event->Correlated Events** from the context menu of the event can be selected. This will open an Event Browser displaying the correlated events.

If events are acknowledged while correlated is checked, all correlated events will be acknowledged. If they are unacknowledged, only the latest of the correlated events will be set to unacknowledged.

Finally, the displayed events can be restricted to only those events that are assigned to objects that are within a subtree of a selected object. This e.g. can be used, to display only events that are assigned to nodes within a certain network. For this, the Event Browser can be opened by selecting the menu entry **Events->View** from the root object of the wanted subtree.

5.9.1 Event Overview

A quick overview about the currently unacknowledged events is provided by the display in the lower right edge of the OpenScape FM Client (see *Figure 6*). This display shows at the first glace whether new events have been received and how important they are.

Figure 6 Overview - Unacknowledged Events

The numbers behind the colore dots display the number of unacknowledged and uncorrelated events of the respective priority as they are currently displayed within the Event Browser (in the example 1 critical, 2 minor and 12 normal events).

Clicking one of the colored dots opens a list that contains all unacknowledged events of the respective priority.

5.9.2 Event Annotation

When you work on the currently registered problems, you will often want to make some annotations for yourself or for your colleagues about certain events. For this you can use the Annotation Browser. In addition the Annotation Browser can be used to keep track of acknowledgements and unacknowledgements done to the respective event: The **Event->Annotate** menu item from the event's context menu has to be selected to open the Annotation Browser (*Figure 7*).

The lower part of the browser is used to create or modify the annotation text.

You can enter the text of your annotation into the text field and click **OK**. The annotation is registered: it will be shown in the text field in the lower section of the Annotation Browser. The background color of the corresponding Acknowledge button in the Event Browser changes to white to indicate that an annotation has been registered for this event.

In order to modify the annotation event, open the Annotation Browser again, enter the new text and click on **OK**; the new text is stored as annotation.

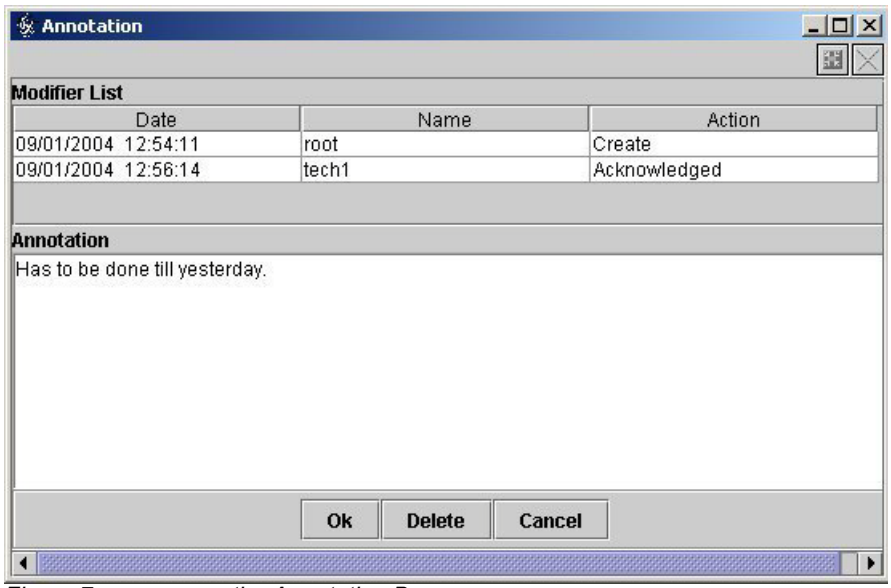


Figure 7 the Annotation Browser

To delete an annotation, just click **Delete**. The background color of the corresponding Acknowledge button changes its color back to default.

The upper part of the browser is used to keep track of the changes that were made to the annotation of the respective event. For each change one entry is added to the **Modifier List**. Each entry consisting of three values which represent the date at which the change occurred, the name of the user that triggered the change, and the actual change type.

Change types are e.g. the creation, deletion or modification of the annotation. But change types can also be the user activated acknowledgement or unacknowledgement of the respective event.

5.10 Message Log

The Message Log (*Figure 8*) lists system messages which are related to the running client. You can display it by selecting **Message Log** from the Quick Navigator of the Submap and Info View Area. The Message Log contains two columns: the severity and the message itself. The very first line of the Message Log indicates where your Client has stored its client log file.

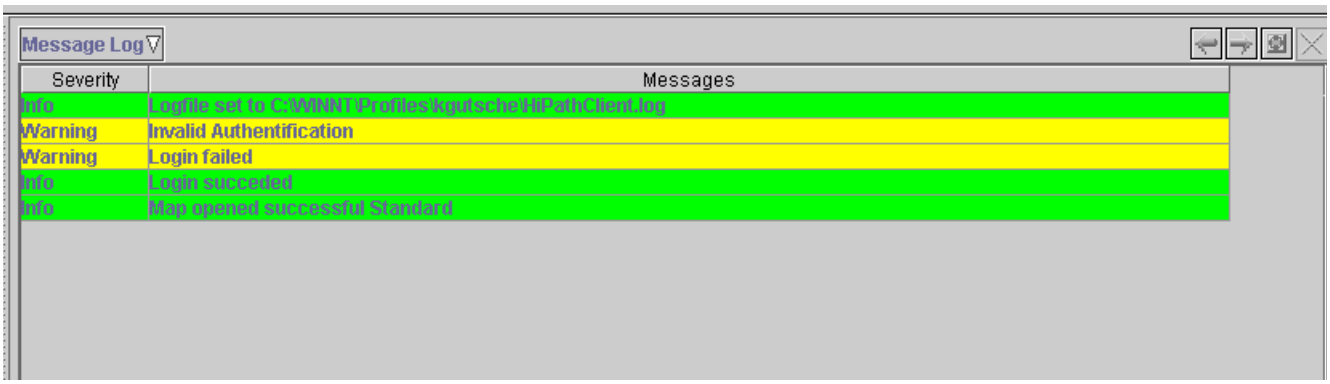


Figure 8 Message Log

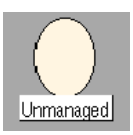
When warnings occur in the Client, the Message Log opens automatically and the message of the warning occur in yellow fields.
Information is printed in green fields, whereas failures cause red list entries.

You can simply accept the messages and close the Message Log by pressing the **Ok** button, or you can clear the Message Log by pressing the **Clear** button.

5.11 Topologies in the Submap and Info View Area

The Topology Viewer is the GUI-component that is responsible for the display of submaps containing network topologies in the Submap and Info View Area, i.e. you will find networks, subnetworks and devices in this view AND the connections between networks or devices. Every object of the network topology is displayed by a symbol. The appearance of a symbol depends on the type and the state of the node it represents. States are represented by colors.

There are eleven different states:



Unmanaged → light brown
This object is still registered in the database, but is not monitored anymore.

Testing → salmon
The status “testing“ means an object is undergoing temporary diagnostic or maintenance procedures.

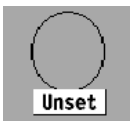
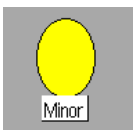
	Disabled → dark brown The status „disabled“ means an object is inactive (although the object may be functioning correctly).
	Restricted → (brown) The status „restricted“ means an object is functioning normally, but it may not be available to all users.
	Unset → gray For this object, no status has been set.
	Normal → green This object is a normal operational state.
	Warning → light blue The object may face a potential problem.
	Minor → yellow The object has a minor problem; this status should not, however, interfere with the normal use of the object.
	Major → orange The object has serious problems; and these problems might also obstruct the normal use of the object.
	Critical → red The object has serious problems and is not working correctly or not working at all.
	Unknown → blue The status of an object cannot be determined.

Table 2 *Status of Objects*

The arrival of certain alarms causes the symbol(s) of the corresponding objects to start blinking in order to notify the administrator that a new event has arrived. Objects can have a compound state which represent the state of all their child objects (status propagation), thus such a symbol will also start flashing, when a new alarm for a child object has occurred. When all events for the object, including all child objects, have been acknowledged (via Event Browser or via the object’s context menu), the object stops blinking. That way you know that blinking icons really represent events that have to be taken care of.

You can select symbols and move them around on the submap with the left mouse button. You can also select multiple symbols at once by pressing and holding the shift-key while you select one object after the other.

The Topology Viewer shows all connections between the network nodes.

5.11.1 User Defined Connections

Sometimes network objects cannot provide any connectivity information related to their communication partner systems. The “User Defined Connections” functionality enables a user to draw such existing connections on the respective submaps.
Select an object and choose **Edit->Link** from its context menu. This object is then marked as source. Select the target object and choose **Edit->Set Target** from its context menu and enter the desired label for the connection. Then press **Create**: the new connection will be displayed immediately.

If both selected objects are located on the same view (i.e. submap or tree), the connection is then drawn between those two objects. If the two objects are located on different submaps, you will be asked, if you want to create the connection on the source map (*Figure 9*), and if you want to create the connection on the target submap. On the submap you choose, the reference symbol will indicate the other end of the connection.

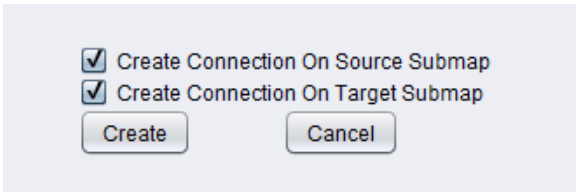


Figure 9 Menu for user defined connection across submap borders

That way you can establish logical links (i.e. create connection symbols) between networks, subnetworks or systems and other devices.
To delete a user specific connection, just choose **Edit->Delete Object** from its context menu.

We now start to work with different maps and submaps. In order to do this there must be IP devices known to the system. Please refer to the *IP Manager Plugin User Guide* to learn how to add the IP devices to your managed network.

5.12 Topology Navigation

As mentioned earlier in this chapter, each managed resource is represented by an object, and objects are represented graphically by one or more symbols on different views (i.e. submaps and trees). A System, for example, is represented by one single object internally, but it can be displayed by many different symbols on different submaps and trees (*Figure 10*, and see *Section 3.2, “Symbols”* and *Section 3.4, “Views”*).

You never access objects directly. Instead, you access them by the use of symbols, which are graphical representations of objects. A symbol is associated with a single object (but, again, one object can be related to more than one symbol). If the term “object” is used in the following text, you can think of it as a symbol displayed by the graphical user interface of the Client.

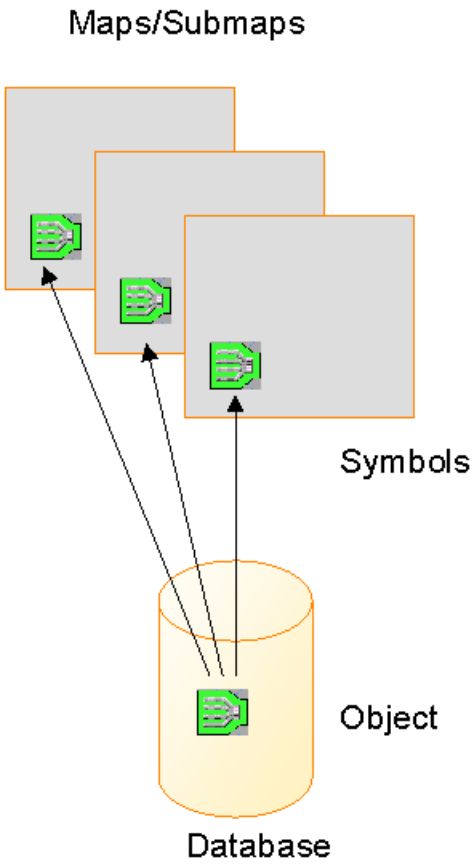


Figure 10 Representation of objects on maps

An object can have a view associated with it. A view contains resources related to the object. In order to navigate from an object to its submap, double-click on the symbol with the left mouse button or choose **Open->Submap** from the object's context menu. A new card is opened in the Submap and Info View Area that shows the submap. You can subsequently move downwards the submap hierarchy by double-clicking on symbols or using the context menu. To navigate upwards, you can use the menu item **Up** from the submap's context menu. In order to show the related subtree on a new card in the Tree View Area, select **Open->Tree**; *Section 5.13, "Navigating Through the System with the Navigation Tree"*.

Of course, the submap hierarchy mirrors the object hierarchy in the database. *Figure 11* shows an example of a hierarchy of three submaps. The top level is the root-submap called "root". It contains a symbol for the user-administration-object. The submap of the User Administration object contains the symbols for rights-administration, group-administration and user-administration (second window). The User Administration object shows all user objects, as shown on the third window.

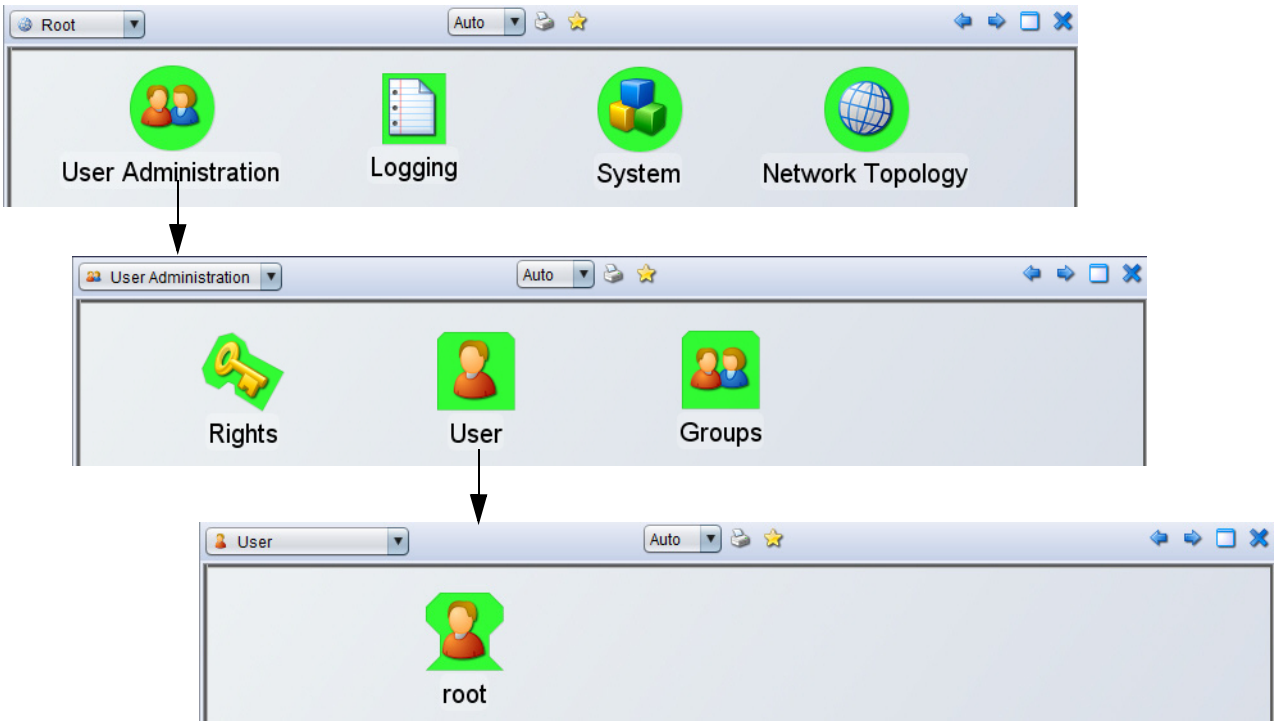


Figure 11 submap hierarchy: User Administration

5.13 Navigating Through the System with the Navigation Tree

In very complex networks it can be confusing to work with a large number of submaps. Sometimes you might be in danger of losing your bearing between all the submaps. The Navigation Tree (Figure 12) will help you to find your way. You see the Navigation Tree in the left section of the main window, the Tree View Area. Every object is represented by its specific symbol and forms a subtree of the tree with its child objects as sub-branches. You might know this principle from the common file-system explorers. The symbol type and label are also location specific. When you have changed the shape and/or bitmap on a submap or in the tree at one location for an object, it will be displayed with this appearance at that specific location in the object hierarchy, but it will be represented with its original shape and bitmap at locations where it has not been modified. The Tree View Controls contains the elements (*Figure 12*) you already know from the Submap and Info View Area: the **Quick Navigator** and the buttons **Forwards**, **Backward**, **Detach**, and **Close**. The Tree View Area follows the same principle as the display area: you can browse in all opened trees (**Forward** and **Backwards** buttons), and you can detach trees (**Detach** button) to have them displayed in a separate window. You can open new trees in new cards which are then available via **Quick Navigator** with the detached tree cards marked in brackets.

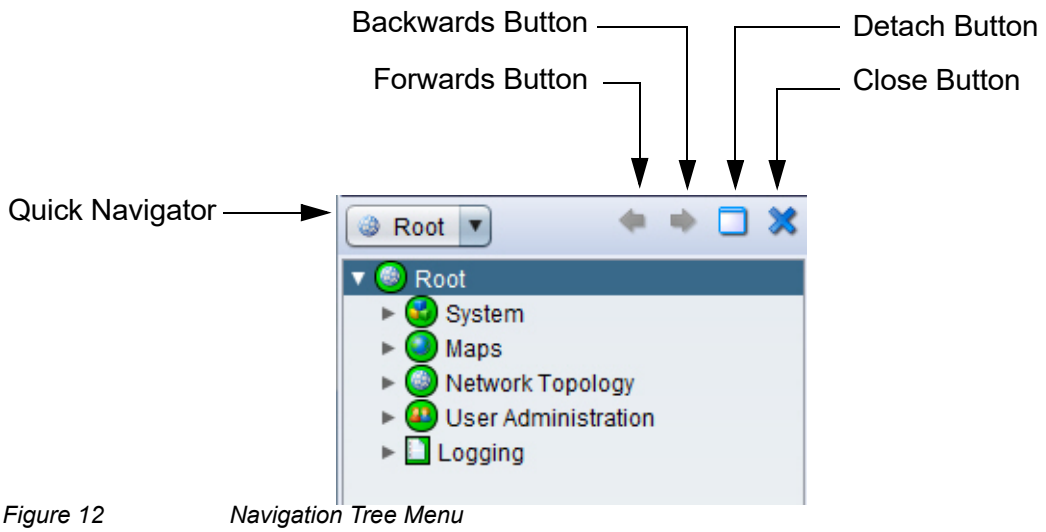


Figure 12 Navigation Tree Menu

You can view all sub-branches of a symbol, i.e. all child objects, with a double click on that symbol. Except for the root submap, every object can also be opened and closed by the help of the switch (“turner” in official Java(TM) terminology) on the branching out of each sub-branch.

A horizontal turner (*Figure 13*) indicates that this sub-branch is closed. You can open it with a simple mouse click on the switch which then turns to vertical orientation. If the object contains any child objects they will be indicated as sub-branches.

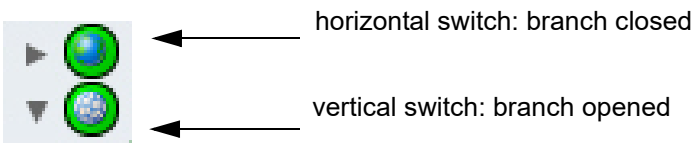


Figure 13 Switch in Navigation Tree

A branch can be opened by a double click on the respective symbol, or by a single click on the switch on the left side of the symbol.

If you want to display/open a submap in the Submap and Info View Area, please use the context menu of the object’s symbol in the Navigation Tree and select **Open->Submap**. You can also have a certain object/ subtree displayed as a new tree: open its context menu and select **Open->Tree**. The selected object will be at the root of the new tree, i.e. you can exclude all other objects from your subtree. Furthermore the name of the new subtree will appear as an item in the Quick Navigator which allows you to switch between all opened tree panels. The currently displayed tree panel is marked as selected.

If you open an object in the Navigation Tree which would contain the object itself in the subtree, i.e. when there is a recursion in the succession of the tree objects, an object’s symbol which has already been opened is marked by an “R” (recursion) in front of its label. It is not possible to open this object one more time, i.e. to extend the tree at that point.

As you have already noticed, in the Tree Views, the symbols offer the same functionalities via context menus as the symbols in the submaps in the Submap and Info View, i.e. you can analyze or configure your network representation by either using the Navigation Tree or the submaps.

5.14 Drag & Drop

As familiar from many operation systems, drag & drop is supported to move or copy objects within the user interface.

To copy an object, the control key has to be pressed while positioning the object.

This can be done within the tree or the submap area. For example, an object can be copied into a submap by opening the submap, selecting the object in the tree, and dragging it into the submap while the control button is pressed.

Whether objects are allowed to be moved or copied, depends on the context.

5.15 Favorites

Some submaps, trees or event browser searches are frequently used. To provide an easy access to these, submaps or configured searches can be saved under the main menu **Favorites**.

They are generated and saved by pressing the button  within the extra control panel, and selecting a name for the new favorite. A menu entry using this name will be added to the **Favorites** main menu.

Three entries are always present:

- **Submap:** This navigates to the home submap.
- **Navigation Tree:** This navigates to the home tree.
- **Events:** This will open the event browser.

Favorites can be deleted by deleting the respective object within the user management in the submap Favorites. The three default entries cannot be deleted.

A submap or tree favorite can be reconfigured by dragging an object onto the respective favorite symbol. The submap of the dragged object will then be used for the favorite. This can also be done with the default entries Submap and Navigation Tree.

Within the Submap View, the submap of an object or the submap itself can be made the Home Submap by selecting the entry **Submap->Set Start Submap** from the context menu of an object or the submap itself.

The favorites are attached either individually to the current user or to an assigned Start View group (see Section 13.1.1).

5.15.1 Autostart

While Favorites provide an easy access to views that are often needed, the Autostart function defines views that are automatically created when the Server is started.

The definition of Autostart views is based on the definition of Favorites. All Favorites that are copied to the submap **Autostart** within the user management will be automatically opened when the Client is started. This will be done in the same order as they are displayed within the navigation tree.

The order within the tree can be modified by selecting an entry and using the key combination Shift + Up/Down arrow.

For users whose Start View (see *Section 14.1.1*) and Autostart has been attached to a group, the definition is done via the corresponding group.

5.15.2 Toolbar Favorites

Selected Favorites can be added to the Toolbar (located above the main menu bar) to provide direct access.

The definition of Toolbar Favorites is based on the definition of Favorites. All Favorites that are copied to the submap **Toolbar** within the user management will be displayed within the Toolbar.

For users whose Start View (see *Section 14.1.1*) and Toolbar has been attached to a group, the definition is done via the corresponding group.

5.16 Object Annotation

Individual comments can be assigned to the OpenScape FM objects.

These are displayed as tooltips on all symbols that represent the corresponding object.

Comments can be configured or modified by selecting **Properties** from the context menu of a symbol for the desired object. The configuration then is done on the page **Comments**.

5.17 Status Explanation

Why an object has its current status can be displayed by choosing **Status Explanation** from the context menu of the corresponding object.

The window that opens contains the reason or base for calculating the current status (see also *Section 13.3*).

If the status is defined by other objects (compound status) and corresponding events the window contains a list of objects and events that are responsible for the current status.

6 Server Configuration

This chapter describes the basic functions to configure the OpenScape FM Server.

These functions are accessed from the main menu bar of the client by selecting the entry **Server->Administration->Server Properties**. This opens the configuration window, which offers the various configuration options on pages described in the following sections.

6.1 Mail Configuration

On the page **Mail Configuration** the connection to the mail server can be defined through which the OpenScape FM sends its mail messages, which, for example, are sent by the Event Correlation Engine or Reporting.

The mail **Protocol** to be used, the **Server** and the desired **Port** can be specified.

The **Mail-Account/User** and **Password** fields define the account and its authorization that the OpenScape FM will use to send its mails.

6.2 Server Process Parameters

On the page Server Process Parameters the resources that are used by the OpenScape FM server can be limited.

- **Maximum Memory**:
  The value specified here limits the size of the main memory occupied by the OpenScape FM Server. If the value is exceeded, the system swaps out to the paging file accordingly.
- **Maximum Log File Size**:
  This value limits the size of the log files.
  If the value is exceeded, a new log file is started, the old one is moved to `name_old`, an existing `name_old` file is moved to `name_older`, and an existing `name_older` file is deleted
- **Enable old TLSv1 and TLSv1.1 protocols**
  If this checkbox is set, the outdated Transport Layer Security protocols are used.
  This is not recommended, but may be necessary when certain combinations of Java versions are used.
  If this checkbox is not set, TLSv2 is used.

6.3 Active Directory Configuration

The OpenScape FM allows users to automatically log on using Active Directory.

On the page **Active Directory Configuration** this function can be activated and the Active Directory Server to be used can be selected.

More on this can be found in *Section 14.1.9*.

6.4 Event Browser

On the page **Event Browser** it is configured which events are to be automatically deleted from the Event Browser.

An event is automatically deleted if the number of existing events exceeds the maximum limit specified in the **Maximum Event Number** field.

The selection menu **Event Deletion Policy** then determines which event is deleted.

- **Oldest**
  The oldest existing event is deleted.
- **Oldest Acknowledged**
  The oldest acknowledged event is deleted. If all events are acknowledged, the oldest existing event is deleted.
- **Oldest Acknowledged By Severity**
  The oldest event among the acknowledged events with the lowest status is deleted. If there are also no more confirmed *Critical* events, the oldest unconfirmed event of the lowest status is deleted.

In addition, the page can be used to specify whether acknowledging an event automatically removes the *In progress* attribute from that event as well.

6.5 Proxies

The page **Proxies** is used to configure connections to IP addresses that are not located in internally monitored networks.

The page consists of a list of addresses and address ranges, to each of which a proxy can be assigned. This proxy is then responsible for connecting the corresponding IP addresses.

The list is evaluated prior to a contact with an IP address in order of the **Index** number, and the first entry that matches the desired IP address is executed.

OpenScape FM usually tries to establish an HTTP (HTTPS) connection first. If this does not succeed, a SOCKS connection is attempted.

Using the buttons placed below, entries can be created, or selected entries can be deleted or moved within the list.

The fields contain:

- **Index**
  The sequence criterion for the evaluation.
- **Network**, **Address** and **Mask**
  These fields define the IP address range for which the respective entry is responsible.
  In the field **Network** or **Address** the host name or the IP address of the corresponding IP node or network and in the field **Mask** the netmask (e.g. `255.255.240.0`) can be entered.
  The OpenScape FM determines the matching counterpart for entered names or IP addresses and automatically transfers them to the **Network** and **Address** fields.

- **Host** and **Port**
  These two fields define the proxy to be used to connect to the appropriate IP addresses.
- **Type**
  Defines the type of connection. This can be done via *SOCKS* or *HTTP*.
  If *Direct* is selected, a direct connection is established without an intermediary proxy. This can be used, for example, to exclude a single node of a defined network from the proxy usage by adding the direct entry to the proxy list prior to defining the network.

6.6 Database Connection

The page **Database Connection** can be used to select a default database, configure it or set up a new default database.

In the menu **Database Connection** one of the currently defined databases can be selected as the default database. The **button** next to it can be used to configure the access account for the selected database.

The button **New Connection** introduces a new database in OpenScape FM.

In addition to an **Object Name** to identify the database definition, under **Driver** one of the database drivers can be selected that are located in the directory
`<OpenScape FM_installation directory>/server/lib/external` or
`<OpenScape FM_installation directory>/server/lib/external`.
Selecting the driver selects the type of database and clicking the **Create** button opens a window in which the access parameters can be set.

6.7 Data Export

On the **Data Export** page it can be specified when old data is to be automatically deleted from the OpenScape FM database.

Distributed over three pages, it can be defined separately for **Event**s, **Status** changes and System Management **Parameter**s how old data has to be to be deleted (**Clean Up Interval**), and how often it will be checked, whether such data exists (**Cleanup Check Interval**).

For system management parameters, it can also be selected for which external agents data is to be exported at all (**Export Parameter Data for Agent**).

6.8 Update

The automatic software update function can be configured on the page **Update**. More on this can be found in *Section G.1*.

6.9 SSL Certificates

On the page SSL Certificate it can be specified whether the server should use HTTPS and which parameters are used to create the self-signed certificates. More on this can be found in *Section 26.2*.

6.10 Info

The page **Info** displays general information about the server

For example, it displays the free and used memory on the server host, the date of server and database creation, the installed version and the installation dates of the previously installed versions.

7 Object and Event Search

The OpenScape FM provides functions to search for objects or events. These are explained in the following sections.

The easy Quick Search (see *Section 7.1*) provides fast access to objects whose properties contain the chosen substring.

In contrast the complex Object Search (see *Section 7.2*) and the Event Search (see *Section 7.3*) allow search requests in which every conceivable object and event property can be used as a filter criteria.

7.1 Object Quick Search

The Quick Search is a search filter that searches for OpenScape FM objects which contain a specific substring within one of their relevant properties. Among the relevant properties are the object label, the unique object name and comments for the object or the annotation for the event.

The Quick Search input field is located within the right half of the Main Menu Bar and a search is started by writing the search string into the textfield and pressing *Return*.

To e.g. perform a Quick Search that should find a specific host or objects with a specific comment, simply parts of the respective host‘s name or the comment have to be typed into the field and *Return* has to be pressed.

More than one entry, separated by space, can be entered into the input field.

Entries that consist of a single character will be ignored by the search to avoid useless search results.

An object matches a search, if at least one relevant object property contains *all* search entries as a substring.

The search is case insensitive and wild cards are not supported. With the exception of upper/lower case, everything will be interpreted exactly as typed.

All objects that match a search will be displayed within a result table. This table shows the **Object** label, the object **Type**, the unique **Internal Object Name** and on how many submaps the respective object is currently **Hidden**. The usual context menus for the table entries can be used to access the individual objects and their functions.

By default the objects are sorted by relevance of their object type, with the most relevant types like IP nodes and IP networks at the top.

7.2 Object Search

The OpenScape FM Desktop provides a dialogue to search for objects that contain a common characteristic. It is possible to:

- Find objects in either the open map or (if the corresponding submap is not yet open) in the object database that contains the search objects.

- Create a search request with particular object and symbol characteristics, such as all users, all ip nodes or all objects with a specific label.
- Open the submap of each found object directly out of the search result list.

Selecting the **Client->Object Search...** menu item opens the 'Object Search' window to perform a global search. If a search should be performed that is restricted to a subtree of the object hierarchy, the menu entry **Open->Object Search...** can be selected from context menu of the root object of the desired subtree. This will open the same window. If an object search has already been performed for a subtree, the same search can be performed again by using the context menu of the subtree‘s root object and the entry **Open->Last Search**.

The tab **Object** in this window is used to perform object searches. When objects should be found, this tab has to be active while the **Search** button is pressed. In this case the name of the current page is also **Object Search**.

There are different object/symbol characteristics which can be used to search for objects contained in the OpenScape FM database:

- **Label:** The map-specific label of the symbol that is displayed on a submap or in a subtree. A search pattern can be entered in this text field. Such a search pattern can be a simple substring or a regular expression. For more details of regular expressions please refer to *Section 18.2.1, “Configuration of Filters”*.
- **Object Name:** The unique database name of the object at the OpenScape FM server. In this text field you can enter an arbitrary substring or a regular expression. For more details of regular expressions please refer to *Section 18.2.1, “Configuration of Filters”*.
- **Technology/Object Type:** The object technology and object type for important objects. Some important objects are classified by their technology and object type. In more details, every important object which has an object type is member of a technology. A common technology is for example “IP” with the object types “Node” (searching for all ip nodes), “Snmp” (searching for all known Snmp agents), “Http” (searching for all known web servers).... Searches can be initiated for all objects of one technology or for one single object type of a technology only. If a technology gets selected in the corresponding choice box, all known object types for the selected technology are listed in the corresponding **Object Type** choice box.
- **Status:** The user can search for objects that are represented by a symbol with the given status like “Normal” or “Minor”.
- **Shape:** The icon shape of the symbol that represents an object on a submap or in a subtree.
- **Bitmap:** The icon bitmap of the symbol which represents an object on a submap or in a subtree.
- **Case sensitive search:** Case-sensitive search for the object names and labels can be switched on or off by this checkbox.
- **Show object names:** If the checkbox is selected, the result list will display the object names of the matching objects. In addition, enclosed in squared brackets, the labels will be displayed. If the checkbox is not selected, only the labels will be displayed.
- **Consider Status Propagation:** If this checkbox is selected, only objects that are able to propagate their status up to the object from which the search was started (or the Root object for a global search) are considered. This can be used to only find those objects that may directly effect the status of the base object of the search.

- **Object Attribute**, **Values:** The field **Object Attribute** can be used to select an object property (see *Section 5.7, “Standard Info Browsers”*). A search will only fiend objects that have the selected property and the value for this property matches the entry in the field **Values**.

If more than one object/symbol characteristic is used, the search engine will join them in an AND-relation, which means that all selected choices must match at the same time. To start an object search, the **Search** button at the bottom of the “Object Search” window has to be pressed while the page **Object** is opened. The **object result** list will display all matching objects immediately. To stop an object search request, the **Stop** button can be pressed. Double clicking on an object in the search result list will open the submap (of the first found parent) where this object is represented.

The buttons **Manage**, **Unmanage** and **Delete** perform the action with the same name from the context menu of all selected objects. This means, the selected objects will be managed, unmanaged or deleted, if this is possible.

The button **Events** will open a window that displays all events for the objects currently in the object list.

With the button **Hide** all selected objects can be hidden, or displayed again with the button **Unhide**. This is identical to the function for symbols described in *Section 12.1, “Hidden Objects”*. In this case the function will not be performed for a single symbol, but for all symbols representing one of the objects selected in the list.

The button **Clear Search** empties all input fields of the tabs **Object** *and* **Event**, removes all check marks and resets all selection menus to their default settings. This function allows the quick initiation of a new search.

7.3 Event Search

The event search uses the same window as the object search described above (see *Section 7.2, “Object Search”*). Like the object search, events can be searched for all objects (by using the main menu entry **Cient->Object Search**) or for objects within the sub hierarchy of an object (by using the menu entry **Open->Object Seach...** from the context menu of this object).

When events should be found, the tab **Event** has to be active while the **Search** button is pressed. In this case the name of the current page is also **Event Search**.

The search will display all events that match the event criteria described below **AND** that are received for an object which matches the object search criteria that are configured on the tab **Object**. Wildcard filters `.*` have to be entered into the object search fields **Label** and **Object Name**, when *all* events should be considered.

The following event criteria can be configured to restrict the found events:

- **Category**, **Source**, **Description**, **Acknowledged**, **Correlated**: These correspond to the respective values displayed in the event browser (see *Section 5.9, “The Event Browser”*). For the fields **Source** and **Description** regular expressions may be used (see *Section 18.2.1, “Configuration of Filters”*).
- **Severity Minimum**. **Severity Maximum**: These define the minimum and maximum severity of the matching events.
- **Time Schedule**: Here a Time Schedule or a Maintenance Filter can be selected during which the matching events have to be received.
- **Start Date**, **End Date**: These restrict the time interval during which the matching events have to be received.

8 Event Actions

Within OpenScape FM, Event Actions are responsible for the creation of the events that are represented in the Event Browser (see *Section 5.9*).

Within the OpenScape FM, three different types of sources exist whose events might trigger Event Actions:

- **External Events** are triggered by general SNMP traps from external sources. The respective MIB has had to be loaded into the OpenScape FM.

  For example: An SNMP trap sent by a printer and stating that a problem has occurred, creates an External Event Action within the OpenScape FM, if configured accordingly.

- **Integrated Events** are triggered by specific SNMP traps that are known and handled by technology specific OpenScape FM plugins.

  For example: Link-Down-Traps are identified and handled by the IP Manager Plugin. If the IP Manager Plugin has been initialized, Integrated Event Actions will be generated, even if the respective MIB has not been loaded into the OpenScape FM.

- **Internal Events** are triggered by the OpenScape FM itself and are used to inform users about facts discovered by the OpenScape FM.

  For example: If a license failure is detected within the OpenScape FM, or if an IP node can no longer be reached by the OpenScape FM, an Internal Event Action is created.

For all three trigger types, **Events** (see *Section 8.1*) and **Automated Actions** to these events (see *Section 8.2*) can be configured within the OpenScape FM.

In some cases, **Manually Triggered Actions** are desired for already listed events. How these are handled and configured is described in *Section 8.3*.

If certain events are to be suppressed for predefined or all objects, the desired ignore can be configured using a simplified procedure (see *Section 8.4*).

8.1 Customization of Events

The Event Customization defines which source events should be handled by the OpenScape FM and how these events should be structured and displayed within the Event Browser.

The created OpenScape FM events are additionally assigned to the object to which they best fit. Usually, this is a child object of the IP node whose IP can be assigned to the event. Therefore, for example, an interface event is usually assigned to the appropriate interface object of the IP node to which the interface belongs.

Hint:
In some cases, events can be assigned to multiple objects. For some technologies, for example, specific events are assigned to a component and the parent IP node.
If such an event is acknowledged, the acknowledgement will affect all objects to which the event has been assigned.

If an event cannot be assigned to a specific child object of an IP node, the child object **Events** is used by default.

The Event Customization provides a list of all possible source events and the option to generate OpenScape FM events based on the values of these events.

By default, the OpenScape FM provides predefined event configurations for the many internal and integrated events.

The customization is handled by two pages. First the Event Customization has to be used to select an event source (see *Section 8.1.1*), and then the Event Configuration Browser can be used to define the individual events for the selected source (see *Section 8.1.2*).

8.1.1 Event Source Customization

The Event Customization is used to select an event source. It can be opened by using the entry **Event Types** from the main menu **Server**.

The configuration window contains a list of the possible sources of events. Each source for internal or integrated events is generally represented by the name of the plugin that generates the events. For external sources (SNMP traps) one entry for each Enterprise MIB definition file that has been loaded into the OpenScape FM is shown.

Double clicking a source from the list or selecting a source and pressing the button **Configure** opens the **Event Configuration Browser** for the selected source.

8.1.2 Event Configuration Browser

The Event Configuration Browser is used to define how source events should be handled, depending on their type, by the OpenScape FM and displayed in the Event Browser. Each Event Configuration Browser is opened for a single source of events as described above.

The browser window contains a table where each row represents one event defined by the respective source. In the case of external events, these are the events defined by the corresponding definition file.

The column **Trap** contains the names of the different source events. All other entries can be reconfigured by using the respective fields, checkboxes and menus on the right hand side of the window.

All text fields may include macros which will be replaced when an event is received. Macros start with a dollar symbol ('$') followed by a string defining the content. The possible macros are shown in the following table.

`$1` - `$99`	value of the respective trap or event variable
`$#`	number of the variables
`$*`	all trap or event variables
`$@`	the time in seconds when the event was generated
`$T`	the system time of the SNMP agent system, when the trap was sent (for external events)
`$x`	the time as a formatted character string, when the event was generated
`$R` or `$r`	the source IP

Table 3 Macros

`$e`	the Enterprise OID of the event
`$h`, `$H`	the hostname / fully qualified hostname, or if unknown, the IP address of the event's source

Table 3 Macros

The field **Message** configures the definition of the string that will be displayed in the column *Description* within the Event Browser. The column **Category** defines the content of the Event Browser column of the same name.

The field **Severity Format** configures the severity of the event which is displayed in the Event Browser's column *Severity*. Here one of the constant values from the selection menu (e.g. '*Normal*', '*Critical*') can be chosen.
For external events: If a ressource key has been defined within the MIB Editor it will be displayed here.

The checkbox **Managed** defines whether respective events should be evaluated or not.

The checkbox **Flash** defines whether the affected object should flash in the OpenScape FM, if a respective unacknowledged event exists within the Event Browser.

The checkbox **Client Massage** defines whether a Client Message should be generated when respective events are created.

Events: Automatic Acknowledgement

The OpenScape FM has a default mechanism that ensures that new incoming events automatically acknowledge all existing 'similar' events. This is to ensure that e.g. traps reporting positive events automatically acknowledge the associated problem traps.

To make this possible, at least one key is automatically assigned internally to each incoming event. Already existing events, with one of these keys, are automatically acknowledged at this moment. The type of these events does not matter.

The fields **Event Type Format** and **Event Type Separator** define, how the key will be generated for the selected event type.

The field **Event Type Format** defines the format template which is used to generate the key. The macros listed above can be included in this template.

If more than one key should be generated for an event, the **Event Type Separator** field can be used to specify a symbol with which the string from the Event Type Format field will be split. All keys generated in this way are associated with the corresponding event in the Event Browser.

Example:

A trap is received and the trap variables $3, $4 and $5 contain the values '100', '15' and '36'.

If the Event Type Format is defined as 'XTrap $3 $4' and no separator was defined, the key for this trap will be: 'XTrap 100 15'.

If the Event Type Format is defined as 'XTrap $3 $4&XTrap $3 $5' and the separator '&' was defined, the keys for this trap will be: 'XTrap 100 15' and 'XTrap 100 36'.

If the current new event should be confirmed in addition to the already existing events with the same key, the fields **Reset Type Format** and **Reset Type Separator** can be used instead. In this case, the fields for the Event Type Format have to remain empty.

Example:

For the trap `Link-Down` the **Event Type Format** `$r:Down:$1` has been defined.

For the trap `Link-Up` the **Reset Type Format** `$r:Down:$1` has been defined.

If a `Link-Down` trap is received for which `$r` will be replaced by `192.168.1.1` and `$1` by `4`, the key `192.168.1.1:Down:4` will be attached to the event generated by the trap.

If at a later time a `Link-Up` trap is received for which `$r` is also replaced by `192.168.1.1` and `$1` also by `4`, then the chosen Reset Type Format generates an identical reset key and the `Link-Down` event of the first trap and the new matching ‚positive' `Link-Up` event will be acknowledged.

If neither an Event Type Format nor a Reset Type Format is assigned for an event type, a default format is used. This format composes the key from the source of the event and the name of the event type. This ensures that events automatically confirm existing events of the same type for the same object.

For many event types useful configurations will be entered automatically during the installation of the OpenScape FM.

Usually these defaults ensure that events with status '*Normal*' acknowledge all content-matching events of the same source with a different severity.

Events: Duplicate Suppression

The fields **Duplicate Format** and **Duplicate Time** can be used to prevent the evaluation of duplicated source events (e.g. to prevent the double evaluation of an event that is send from two monitoring sources).

The *Duplicate Format* is handled like the keys described above. Based on the format, a *Duplicate Key* will be created for each incoming event for which a *Duplicate Format* has been defined. It will then be checked whether another event of the same type that has the same *Duplicate Key* has been received within the time interval defined by the *Duplicate Time*. If this is the case, the event gets tagged as a duplicate and will be ignored by default.

Note:
Incoming events are handled by the Event Correlation Engine Plugin. Within the ECE the handling of events marked as Duplicates can be modified (see separate *Event Correlation Plugin User Guide*).

Pressing the buttons below the table will **Reload** the content of the table, **Close** the window or will open the **Event Action** configuration for the current event (see *Section 8.2*).

8.2 Automated Reactions for Events

An Automated Reaction for an incoming event is a predefined action that is performed whenever an event of a specific type is received for a specific object during a defined time interval.

Important Note:
The Event Correlation Engine Plugin is needed for this feature.

Fully configurable Automated Reactions can be assigned to all events that are handled by the OpenScape FM (see *Section 8.1*). Automated Reactions are therefore a mechanism that allows the OpenScape FM to instantly react to incoming events in a way that is fully under control of the OpenScape FM administrators.

The following subsections explain, how actions can be assigned to a combination of events, objects and time intervals do define an Automated Reaction (see *Section 8.2.1*), and how the object filters (see *Section 8.2.2*), time filters (see *Section 8.2.3*) and actions (see *Section 8.2.4*) that are combined can be configured.

For the special case that certain events should only be ignored, a simplified configuration procedure is described in Section 8.4.

8.2.1 Defining Automated Reactions

If configured accordingly, an Automated Reaction is performed, when an event of a specific event type is received for a defined object within a defined time interval.

Important Note:
For this functionality the Event Correlation Engine Plugin is required.

The configuration of Automated Reactions for events is handled within a window that is opened by using the entry **Event Filter** from the main menu **Server**.

For individual event types, a similar configuration window can be opened by using the entry **Configure** from the context menu of a symbol that represents the respective event type. Such symbols are e.g. available during the configuration of event types (see *Section 8.1*).

Note:
Besides the missing selection panel for event types, the configuration windows are functionally identical.

The configuration limited to an individual event type can also be opened from the context menu of an event. Here the entry **Event->Event Filter** opens the configuration for the respective event type.

On the page **Event Filter Configuration**, actions can be assigned to combinations of event types, object filters and time filters.

The page contains a list that displays one configured Automated Reaction per row. Each of these rows displays the **Event Type** and the *Object*, *Time Schedule* and *Action* that is connected to this Event Type.

New entries can be created by pressing the button **+**. Selected entries can be deleted by pressing the button **-**.

Selected entries can be modified by changing the configuration parameters with the selection elements on the right hand side:

- The panel **Event Types** displays a selection tree with the internal, integrated and external event types known by the OpenScape FM (see *Section 8.1*). They are ordered by their source. Event Types that are currently not monitored are displayed with the status *Unmanaged*.
- The menu **Filter** is used to select the objects that should be assigned to the Automated Reaction. It contains two entries for each Object Filter that has been defined (see *Section 8.2.2*). The respective entries show either a green check mark or a red x in front of the filter‘s label. The check mark entry represents objects that match the filter, the X entry all objects that do not match the filter.

- The menu **Time Schedule** is used to select the time intervals during which the Automated Reaction should be performed. It contains two entries for each Time Filter that has been defined (see *Section 8.2.3*). The respective entries show either a green check mark or a red x in front of the filter‘s label. The check mark entry represents points in time that match the filter, the X entry all other points in time.
- The menu **Actions** is used to select the Action that should be performed by the Automated Reaction. It contains entries for all predefined actions that can be performed as a reaction (see *Section 8.2.4*). The selected Action will be performed, if an event of the selected event type is received and the selected filters are matching.

8.2.2 Defining Object Filters for Reactions

Object Filters are used to define sets of objects for which Automated Reactions should be performed.

Object Filters are automatically created and represented as Event Correlation Engine (ECE) nodes. Depending on the filter type, the nodes are of the type *IP Pattern Filter* or *Object Filter* (see *Event Correlation Engine Plugin User Guide*).

Both types of nodes provide two outputs. One for matching objects and one for not matching objects. Both outputs can be assigned to Automated Reactions. It is therefore possible to create ‚inverted‘ filters.

The configuration of Automated Reactions for events is handled within a window that is opened by using the entry **Event Filter Configuration** from the main menu **Server**. The page **Filter** within this window is used to configure these filters.

The page contains a list of the currently defined Object Filters. Each represented in a separate row.

The button **Create Filter** (**+**) can be used to create a new filter. The type of this filter can be selected from the opened window:

- **Object Filter** can be used to create a new filter that filters for a set of manually defined objects. The filter looks for objects that are selected from a navigation tree or for objects with specific properties and is represented by an ECE node of type *Object Filter*.
- **IP Pattern Filter** can be used to create a new filter that filters for IP nodes that are located within a selected subnetwork. It is represented by an ECE node of type *IP Pattern Filter*.

Depending on the filter type, the button **Configure** can be used to change the objects and object properties or the subnetworks that are assigned to the selected filter.

The buttons **Changes the label of the object** and **Delete Object** can be used to rename or delete the selected filter. The name of a filter is used to represent it within the filter selection menu.

8.2.3 Defining Time Filters for Reactions

Time Filters are used to define time intervals during which Automated Reactions should be performed.

Time Filters are automatically created and represented as Event Correlation Engine (ECE) nodes of the type *Time Schedule* (see *Event Correlation Engine Plugin User Guide*).

The node type provides two outputs that are relevant for the Time Filter. One is for events that are received during the defined time intervals, and one for the other events. Both outputs can be assigned to Automated Reactions. If e.g. a Time Filter has been defined to check for support hours, it can also be used for specific reactions that should be performed when no immediate support is currently available (like informing a technician via phone about very important events).

The configuration of Automated Reactions for events is handled within a window that is opened by using the entry **Event Filter Configuration** from the main menu **Server**. The page **Time Schedule** within this window is used to configure these filters.

The page contains a list of the currently defined Time Filters. Each of these filters is represented by a separate row.

The button **Create Time Schedule** (**+**) can be used to create a new filter that filters for a set of manually defined time intervals. The filter checks for points in time that are within the defined intervals and is represented by an ECE node of type *Time Schedule*.

The time schedules are represented by objects within the container **Root->System->Server->Filter Manager**. If a filter object is displayed as Disabled (dark brown), the filter does not apply at the current moment.

The button **Configure** can be used to change the time intervals that are assigned to the selected filter.

The buttons **Changes the label of the object** and **Delete Object** can be used to rename or delete the selected filter. The name of a filter is used to represent it within the filter selection menu.

8.2.4 Defining Actions for Reactions

Actions define the reactions that are performed, if Automated Reactions are triggered by incoming events.

The actions that are available within the Automated Reaction configuration (see *Section 8.2.1*) can be defined and configured within the Event Correlation Engine (see separate *Event Correlation Engine Plugin User Guide*).

Within the ECE, existing actions can be modified and new actions can be created.

The ECE module responsible for the handling of actions for Automated Reactions is located within the Tier Filter node labeled *Event Reactions* that can be found within the container *Event Source* on the ECE main submap.

The node *Event Reactions* holds the *Action* container in which each output object represents one possible action. The label of these output objects are used as the action‘s names. ECE nodes that perform the desired action can be connected to the respective output objects.

New output objects and therefore new actions can be generated by using the entry **New->Action** from the context menu of the *Event Reactions* node.

The configuration of Actions can also be handled within a window that is opened by using the entry **Event Filter Configuration** from the main menu **Server**. The page **Actions** within this window is used to configure these actions.

The page contains a list of the currently defined Actions. Each represented in a separate row.

The button **Create Action** (**+**) can be used to create a new action. While the method described above allows the creation of any type of action, the window opened by the button can be used to create some often needed generic actions:

- **Debugger** can be used to create a new action based on an ECE node of type *Debugger*. This node type collects incoming events and is mainly used for debugging purposes.
- **External Action** can be used to create a new action based on an ECE node of type *External Action*. This node type generated emails, SMS messages or starts external executables when an event is received.
- **Modifier** can be used to create a new action based on an ECE node of type *Modifier*. With this node type incoming events can be modified. E.g their status can be changed or they can be acknowledged.

The button **Configure** can be used to modify the configuration of the node that represents the respective action.

The buttons **Changes the label of the object** and **Delete Object** can be used to rename or delete the selected action. The name of a action is used to represent it within the action selection menu.

8.3 Manually Triggered Reactions for Events

In addition to the automated reactions described above (see *Section 8.2*), it is sometimes desired to perform certain repetitive reactions for events already present in the Event Browser (see *Section 5.9*).

For example, it may be desired that a technician should be contacted manually for individual events, or that individual events should be handled in a specific way.

For such cases, the Event Correlation Engine offers very extensive possibilities with the Event Menu Entry node provided especially for this task (see separate *Event Correlation Engine Plugin User Guide*).

Each of these nodes extends the context menus of all event objects by an additional menu entry. If one of these entries is selected, the freely configurable workflow associated with the respective node is triggered and filled with the data of the associated event. Every action possible within the Event Correlation Engine can thus be connected to the menu entries of the events and executed with their specific values.

8.4 Ignore Events (Simplified Procedure)

Section 8.2 ff. describes how incoming events can be automatically handled by the OpenScape FM.

This procedure is highly flexible, but also involves a relatively complex configuration.

For the special case that certain events should always be ignored in the future, OpenScape FM offers a significantly simplified procedure that can be triggered from the context menu of a corresponding event:

- The entry **Event->Ignore** activates or deactivates the ignoring of the event type of the selected event for the object to which the event is assigned.
- The entry **Event->Ignore Global** respectively activates or deactivates the ignoring of the event type for all objects.

In both cases, a checkmark behind the entry indicates that a general or object-specific filter exists for the selected event type.

Note:
If the object to which the event is assigned has been removed from the OpenScape FM, the check mark behind **Ignore** is always displayed, since all events are ignored for a removed object.

The easiest way to access existing events is through the event browser (see *Section 5.9*).

If the configured Ignore for an object or all objects is to be undone or replaced by another action, this can be achieved as usual by using the methods described in *Section 8.2.1*.

If the entry **Event->Event Filter** is selected from the context menu of an event, the window described in *Section 8.2.1* also opens. In this case, however, the content is restricted to the appropriate event type.

9 Display of Tray Bar Icons

The Windows platform provides a system tray bar which can be used to display icons which monitor the behavior of system programs or can be used for quick access to functions of these system programs.

The OpenScape FM Desktop contains its own tray bar monitoring which offers the following functionalities:

- Monitoring the status of *one* arbitrary OpenScape FM database object per user.
- Notification (playing a sound, status flashing) of the status changes for the monitoring object.
- Direct start of the OpenScape FM Client with an automatically login.

The tray bar monitoring program displays the state of one defined OpenScape FM object (monitoring object) via a tray bar icon.

The Tray Bar symbol is displayed for the configured monitoring object as soon as the Desktop Client is started. The following symbols can be displayed depending on the status of the monitored object:

Appearance	Category	Status
	Alarm	Critical → red
	Alarm	Major → orange
	Warning	Minor → yellow
	Warning	Warning → light blue
	Ok	Normal → green
	Ok	Disabled → dark brown
	Ok	Restricted → brown
	Ok	Testing → salmon
	Ok	Unknown → blue
	Ok	Unmanaged → light brown
	Ok	Unset → light grey

An arbitrary OpenScape FM object is defined for monitoring via the Tray Bar symbol by selecting the entry **Edit->Set Monitor Object** from its context menu. This ends the monitoring of the object selected so far and starts the monitoring of the current object via the tray bar symbol.

If no object is assigned by the user, by default the user's *Personal View* container is monitored.

The Tray Bar symbol is created automatically when a Desktop Client is opened and by default is closed when the Client is closed.

If in the general preferences (entry **Client->Preferences** of the main menu bar) the check mark is activated in front of **Close to system tray**, then closing the client does not end it, but only makes it invisible while the system tray icon remains.

Using the input field **Alarm sound for monitored object** and the corresponding **Browse** button, a sound file can be selected via a file browser. This sound file is played whenever the status of the object monitored by the Tray Bar deteriorates, e.g. changes from *Normal* to *Minor* or from *Warning* to *Critical*.

The Tray Bar symbol itself offers three functions via menu entries:

- **OpenScape FM Fault Management Client** opens the Desktop Client associated with the symbol.
- **Details** shows an overview window with the current configuration.
- **Exit** closes the symbol and exits the associated Desktop Client.

10 Printing

The Desktop offers print options from two different contexts: you can print the currently displayed submap or you can use the list browser print option. Both functions are invoked with the print button which is displayed in the context dependent Extra Controls area of the Submap and Info View Area.

Figure 14 the Print button

10.1 Printing from Submap

When you use the print option in the submap context, this button opens the standard print menu of your operating system, where you can select the printing parameters.

10.2 Printing from Info Browser

If the print option is invoked from a Info Browser, several more specialized options are available.

In order to print out data from list browsers, e.g. the Alarm Browser or the Discovery Browser, use the print button in the upper part of the Info browser. The print menu (*Figure 15*) which opens offers several configuration parameters for your printout.

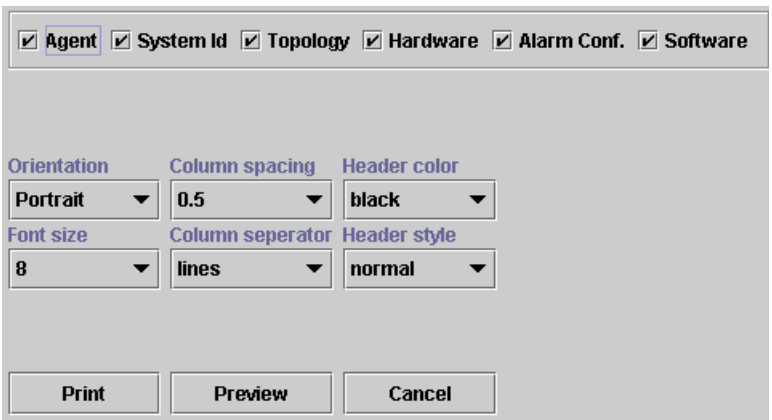


Figure 15 Print Menu for a Info Browser

In the first line, you can choose which columns you want to have printed. The center part of the menu lets you configure the appearance of the printout, e.g. the column spacing or color of the header. You can decide if you want to see a preview first – use the **Preview** button for that. *Figure 16* shows the preview dialogue for the “Object Properties”. Since your printout can comprise more than one page the actual page and the page value will be

shown in the right corner of the preview dialogue. Additionally you can browse through the different pages via the arrows in the middle of the dialogs toolbar. Via the zoom selector you can select the display size of your document – choose between **10%**, **25%**, **50%**, **75%**, or **100%**.

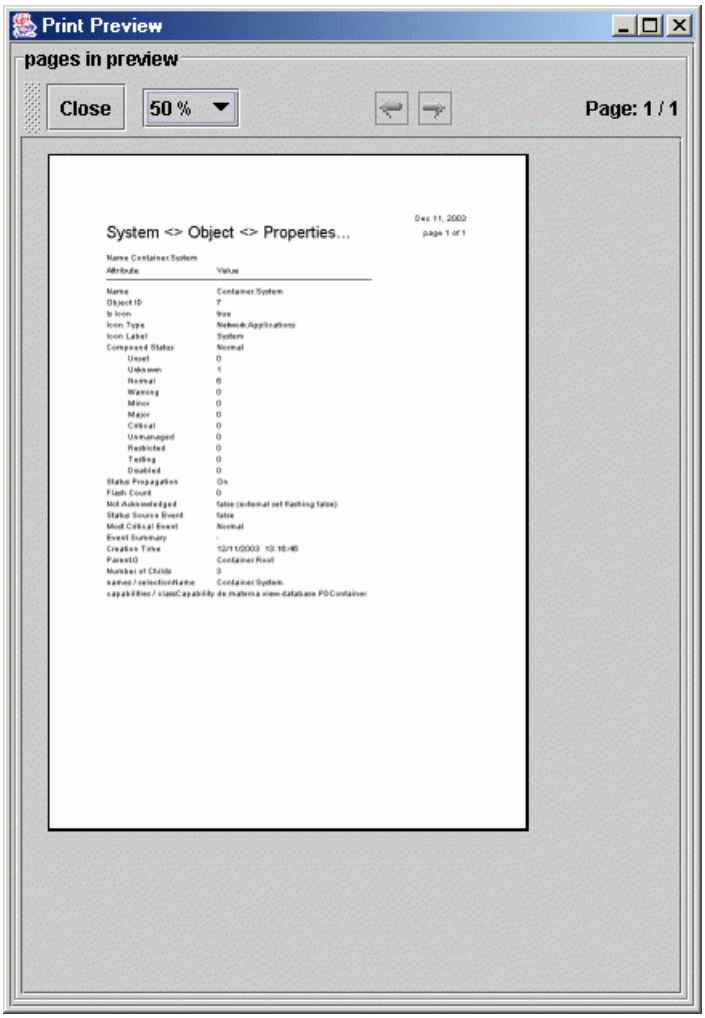


Figure 16 Print Preview, here: Object Properties

If you want to print the data as shown in the preview, press **Print** in the print menu (*Figure 15*). To use this functionality, a printer has to be configured for your computer.

11 User Sessions

User sessions are used to monitor which users are active. This is an internal process. The sessions are used to control the access rights (*Chapter 15, "Access Rights"*), user language and for health monitoring of the client connection.

11.1 User Session Concepts

When a user starts the Client, a session will be established between the client and the server. This session is identified through a session Id.

11.2 Login

The session Id is assigned at login time and will remain the same until the user will logs out. There are mechanisms that will monitor the connection of the client to the server. If a client stops unexpectedly the server will discover this automatically and will free resources claimed by this client.

When you log in to the server, your login name and password are required. The initial password can be set by an administrator during user creation, *Chapter 14, "User and Group Administration"*.

If the administrator has selected **Cleared Until First Login** at the point of time, when the user was created, the user will have to set a password when he/she logs in for the first time. The other two parameters which exert an influence on the duration time of a password are **Min time** and **Max time** which can also be set by the system administrator. *Chapter 14, "User and Group Administration"* for details.

11.3 Time Controlled Login/Logout

If a user is time restricted she will only be able to login during the time intervals defined for the user. In addition the user will be automatically logged out, when the defined time interval is exceeded. If auto login is defined for the user and if an automatic logout was performed, the user will automatically be reconnected at the next allowed login time.

More about time controlled login/logout can be found in *Section 14.1.3, "Time Controlled Login/Logout"*.

11.4 Logout

In order to log out, select **Client->Logout** from the main menu. When a user logs out the session is stopped normally.

User Sessions

Logout

12 Creating Personal Views

You do not have to work only with the views (i.e. submaps and (sub)trees) that have been created automatically by the applications. The Desktop allows to create your own personal views.

Object Container

In the context menu of each user-created symbol and of the root symbol you find in the menu item **New->Object Container**.

You can use this item to establish your own submap hierarchy. Start on the root submap - it's only here, where you can create your first object, the root of your personal submap tree. Usually, you cannot create objects on application-created submaps. In the following process, use **New->Object Container** when you want to add a completely new object to the database, and **Copy/Paste** when you want to insert an already existing object on a different submap. You can create new objects and add existing objects to create your personal view of your network, e.g. include only some systems that you want to have an eye on. You can even choose a personal background for your personal submaps (*Section 5.8, "Selecting a Background Image"*). For each user-created object you can choose if you want it to "forward" its status or not: use the check box **Status Propagation** for that.

When you change the appearance of a submap this will immediately be visible in all maps, to all users who are logged in and work with this submap.

Virtual Container

While the objects must be added manually to the Object Containers described above, Virtual Containers have the ability to dynamically determine their contents during the runtime of the OpenScape FM. This is done according to rules assigned to the respective container.

Virtual Containers can be created using the menu item **New->Virtual Container**.

More about Virtual Containers can be found in the separate *Control Center User Guide* starting with chapter 4.

12.1 Hidden Objects

The appearance of views can be modified by hiding symbols or objects. Hiding means that the respective symbols will no longer be displayed. Events for hidden objects will still be received as usual and they will be displayed within the Event Browser.

Hidden symbols can be shown again.

The following actions are available:

- **Hide:**
  - Symbols can be hidden be using the symbol's context menu (**Edit->Hide**).
  - All symbols representing the same object can be set to hidden by using button **Hide** for the object in the Object Search browser.

- **Unhide:**
  - All symbols hidden on a submap can be made visible by using the submap's context menu (**Edit->Show Hidden Objects**) or by using the same entry within the context menu of an object holding the submap.
  - All symbols representing the same object can be made visible by using the Object Search browser (button **Unhide**).

13 Symbols and Status Display

Objects usually represent actually existing (virtual) objects that are monitored by the OpenScape FM. These objects are represented by Symbols in the views (submaps and/or trees). It is common that an object is represented by several symbols. For example, a host that has several interfaces can be represented by symbols in different networks. Regardless for which of these symbols a function is triggered, it always affects the same internal host object.

In general, the **Shape** and the **Image** of a symbol describe the type of the represented object, the **Label** below identifies the object itself, and the **Color** of the symbol shows the current status of the object.

The first sections of this chapter deal with the appearance of the symbols:

- *Section 13.1* shows some standard symbol optics that the OpenScape FM Desktop automatically assigns to the symbols that represent a specific object type.
- *Section 13.2* shows how the symbol type and label can be individually adjusted for each symbol if required.

The following sections describe how the status of objects can be calculated, how the calculation can be configured, and how the status of objects and further information are displayed by the symbols:

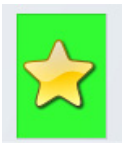
- *Section 13.3* describes how the status of an object is calculated and how the calculation method can be adjusted for individual objects.
- *Section 13.4* specifies how the distribution of the status of the child objects of a symbol is represented.
- *Section 13.5* shows how, in addition to the object status, the reachability of an object is also shown.
- *Section 13.6* shows how to add an information window to symbols on their submap.

13.1 Symbol Optics

This section gives a brief overview of the symbols used by the desktop by default. The description of the plugin specific icons can be found in the respective manuals.

13.1.1 Basic Symbols

- User symbol 
- Group symbol 

- Rights symbol 
- Domain symbol 
- Favorite symbol 

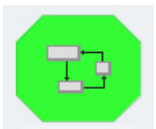
13.1.2 Topology Symbols

- Topology Network symbol 
- Topology Subnetwork symbol 
- Meta edge symbol (tree view) 

13.1.3 Logging Symbols

- Logging symbol 
- Log File View symbol 

13.1.4 System Symbols

- System symbol 
- Help symbol 
- Startup Manger Process 
- License Manager Feature symbol 

13.1.5 Map Symbols

- Map Manager 
- Map symbol 

13.2 Symbol Configuration

The appearance of each symbol can be changed individually using the corresponding context menu. The shape, image and label of each symbol can be configured individually.

Hint:
For system symbols, however, this is only recommended to a limited extent, as it could make it more difficult to identify the symbols.

The configuration of a symbol can be initiated using the context menu entry **Properties** and performed on the page **Symbol->Symbol Properties**.

The configuration window consists of three identically structured sections in which the **Shape** and **Image** can be defined in the top line and the **Label** in the bottom line.

When the window is opened, the currently displayed labels are shown. If a label is determined by a macro, it is displayed in light grey instead of black to indicate this fact.
If the corresponding field is clicked, the macro itself is displayed.

The lower of the three sections defines all symbols representing the selected object, the middle section defines all symbols representing the selected object in the current map, and the upper section defines only the symbol from whose context menu the function was called.

If the **OK** or **Apply** button is pressed, all lines whose checkbox is checked are evaluated.

If a configuration is to be reset to the default setting, an empty entry with a checked checkbox must be entered.

More specific configurations may overwrite the more general ones. For example, if the first and last section is entered, the current symbol is displayed according to the upper section, and all other symbols for the same object according to the lower section.

If only the label should be changed, this can also be done simplified by selecting the symbol in a submap or in the tree and pressing the F2 key, or by using the entry **Rename** from the context menu of the symbol.

13.3 Status Calculation

The status of an object usually describes the current state of the object. If, for example, there are no problems, the status is *Normal*. If there are critical problems, the status is *Critical*. The current status is indicated by the color of the symbols that represent the respective object. The meaning of the individual colors is described in *Section 5.11*.

To get a hint why a symbol has its current status, the menu entry **Status Explanation** can be selected from the context menu of the symbol. This will open an information window which explains why the symbol is in its current status. If the status is defined by other objects (compound status) and corresponding events the window contains a list of objects and events that are responsible for the current status (see *Section 5.17*).

In general, the status of object symbols is determined by the worst unacknowledged event assigned to the object. The status of object container symbols is derived from the worst status of a child object.

But it is possible to define the method of status calculation for individual objects. It can also be defined how the status of an object should affect other objects. The configuration page for the status can be opened by using the entry **Properties...** from the context menu of the respective object. It is located on the tab **Propagation** which consists of four tabs.

Parents:

The tab **Parents** contains a list of all objects to which the current object may pass its status. The list results from the current structure of the object tree.

The column **Object** identifies the respective parent node.

The column **Propagation Weight** defines the absolute weight of the object that should be used for this parent. A value of 0 ignores the object.

The column **Propagation Type** defines whether only the object itself (**Object**) or all child objects (**Hierarchical**) should be used for the calculation.

Children:

The tab **Children** corresponds to the tab Parents. In this case all objects are listed which may pass their status to the current object.

Status Calculation:

On the tab **Status Calculation** in the selection menu **Status Calculation Method** it can be defined how the status of the current object will be determined.

Basically, there are three different approaches to calculate the object status:

1. The status is directly assigned to the object by the OpenScape FM. All other influences on the object are ignored.

   For this method the entry **Object** has to be selected.

   This method is e.g. automatically set by the OpenScape FM for IP nodes up to and including OpenScape FM V10 that cannot be reached. If they are reachable at a later time, the method automatically changes to *Maximum of the child objects*. In general, the *Object* method should not be set manually.

2. The status is determined by the highest status of the unacknowledged events that are directly assigned to the object.

   For this method the entry **Event** has to be selected.

3. The status is calculated from the status of the object‘s child objects.

   In this case the following methods exist:

   - **Maximum of child objects**, **Minimum of child objects**: The status is identical to the highest/lowest status of all child objects that have a Propagation Weight unequal 0.
   - **Absolute Threshold**, **Percental Threshold**: The status is calculated from the absolute or percental distribution of the child objects status values. How the status is calculated in detail is defined on the tab **Thresholds**.
   - **Derived**: The status is calculated with the default method. This takes into account the number of child objects in status *Normal* and the number of child objects in a defective status (*Critical*, *Major*, *Minor* or *Warning*).

     The status is calculated as follows:

     - **Unknown**: no object defective, no object *Normal*.
     - **Normal**: no object defective, at least one object *Normal*.
     - **Warning**: exactly one object defective, at least two objects *Normal*.
     - **Minor**: at least two objects defective, at least two objects *Normal*.
     - **Major**: at least one object defective, exactly one object *Normal*.
     - **Critical**: at least one object defective, no object *Normal*.

Thresholds:

The tab **Thresholds** defines how the status will be calculated if the **Status Aggregation** uses the absolute or percental threshold.

Each line on this tab defines a rule for the respective **Status**. The evaluation is handled in order defined by the **Evaluation Priority**, beginning with the highest priority.

A rule is a match, if at least as many objects are in the respective or a higher status as requested by the column **Threshold**. Depending on the selected Status Aggregation, the weighted absolute or percental values will be used for the calculation. The first matching rule defines the status of the object.

The column **Current Value** displays, how many weighted objects are currently in the respective status (absolute or percental).

Example (Status Aggregation: Percental Threshold, Default-Settings for the Thresholds):

- The object gets the status *Critical*, if at least 5% (weighted) of the child objects are in status *Critical*.
- The object gets the status *Major*, if at least 10% (weighted) of the child objects are in status *Major* or *Critical*.
- The object gets the status *Minor*, if at least 20% (weighted) of the child objects are in status *Minor*, *Major* or *Critical*.
- ...

By using **Thresholds**, it is possible to assign a status to an object that is neither assigned to an event nor to a child object. If e.g. the status *Critical* is set to 80% and the status *Major* to 60%, the status *Major* will be used if 60% of the child objects are in the status *Critical*. This enables flexible reactions for partial failures of (redundant) child objects.

13.4 Compound Status Indicator

The color of every symbol on a map provides information about the status of the underlying object. This can be a single status or a compound status.

A single status represents the status of this distinct system. For example, an object will be displayed as critical when a critical event exists for the object.

A compound status is used, when the status of the object is calculated from the status of the objects located on the submap. For example, a network object will be displayed as critical when a system located in this network has the status critical.

How the status should be calculated can be individually configured for each object (see *Section 13.3, “Status Calculation”*).

If child objects of an object are not all in the same state, the **Compound Status Indicator** appears next to the submap icon (*Figure 17*). It shows roughly the percentage of objects with the indicated system states. This relation refers to the entire object subtree in the object hierarchy, not only to the next level below the specific icon. For example, a **Compound Status Indicator** like the one in *Figure 17* shows that in the network a certain percentage of all managed objects within the subnet are in status “critical” – in this case about half of all managed objects.

Hint:
The Compound Status Indicator considers only symbols for objects that are located in the subtree of the object to which the indicator is attached. Reference Symbols (see *Section 16.2.2*) represent objects that are not located on the displayed submap. They will therefore *not* be considered when the Compound Status Indicator is calculated. Reference Symbols can easily be identified by the << >> brackets enclosing their label.

Figure 17 Network symbol with compound status indicator

For example, the Compound Status Indicator of the “User” symbol shows how many users have been logged in:

- Green user icons symbolize users who were logged in.
- Blue user icons symbolize users who were not logged in.
- Light blue icons show that the corresponding user has tried to log in but failed.

13.5 Reachability Status

In addition to the status described above (see *Section 13.3*), each monitored object has an **Reachability Status**. This indicates whether the object itself is currently reachable or not.

For example, if an object does not respond to a ping within a specified time interval, it cannot be reached.

If the status differs from the Reachability Status, the Reachability Status is displayed as a colored dot at the upper right of the object symbol (see *Figure 18*).

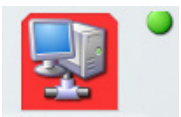


Figure 18 Reachability Status

The host in the figure above has a faulty service and is therefore displayed in *Critical* status.

However, since the host itself reacts to pings, it has the Reachability Status *Normal*.

Inversely, a switched-off PC without any other reported problems would be displayed with the status *Normal* and the Reachability Status *Critical*.

Since the Reachability Status is calculated individually for the objects of some technologies, it is advisable to refer to the Status Explanation of the respective object in case of doubt (see *Section 5.17*).

A network is considered unreachable if none of the Numbered Router Interfaces known within the network can be reached. In this case, the Reachability Status of the network is *Critical*. In addition, in this case the OpenScape FM assumes that the contained objects are unreachable due to the network. These objects are therefore *not* also set to the Reachability Status *Critical*, but to the Reachability Status *Unknown* until at least one of the numbered interfaces of the network can be reached again.

13.6 Symbols as Windows

If the Control Center Plugin is initialized and licensed, symbols can be extended with an information window that can be displayed on the same submap as the symbol.

If such a window has been configured for a symbol, it can be opened by a window icon, which is displayed in the upper right corner of the symbol. The window can then be moved and resized using the usual methods.

Whether a window should be created and what information it should contain can be determined via the entry **Properties...** from the context menu of the corresponding symbol. The respective configuration can be found on the page **Symbol as Window** and the contents of the window can be defined in the selection menu of the same name.

If a window content is defined, the appearance of the symbol changes according to the selection.

If the window for the corresponding symbol should no longer be displayed, this can be achieved by selecting **Reset Symbol Representation**.

14 User and Group Administration

The OpenScape FM Desktop supports user administration where all user-related tasks like changing a password, adding a new user, creating a user group or assigning specific rights to users/user groups can be performed. This chapter will discuss the creation and administration of users and user groups. The assignment and configuration of rights will be described in *Chapter 15, “Access Rights”*.

The starting point of the user and user group administration is the “User Administration“ symbol on the root view. On the view of the „User Administration“ the “User“, “Groups“ and the “Rights“ symbols are accessible. The “Users“ symbol is used for the administration of single users. The “Groups“ symbol is the starting point of the user groups administration. The “Rights“ symbol is the container object for all existing rights.

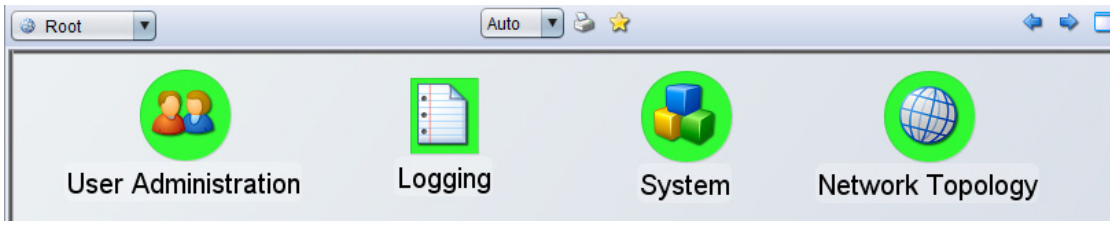


Figure 19 the root submap

The User Administration can be opened by using the main menu entry **Server->Administration->User Administration...** .

14.1 User

The user symbols are located at the **User** view. For each user defined in the system, one symbol will be displayed. By using the context menu of the respective user symbol, the configuration of this user can be changed or the user can be removed from the system.

In detail, the following operations are available:

- **Password:**
  - **Change...:** Here the password for the user can be changed. Without administrator rights, this option is only available for the user that is currently logged in. A user can also change its password over the main menu item **Client->Change Password...**.
  - **Force Change / Do not force Change:** An administrator with the respective rights it can be (un)forced that a user has to set a new password at the next login.
  - **Delete:** As an administrator the password of the user can be deleted; The user will be forced to set a new password during the next login.
- **Lock user/Unlock User:** This option can be used to temporarily disable or re-enable a user account. Administrator rights are required to perform this operation.

- **Configure:**
  - **Information:** This page contains the user data which was entered when the user was created; The user information can be changed here. This page mainly corresponds to the page in which new users can be defined (see *Section 14.1.1*).
  - **Object Rights:** This page can be used to assign object rights to the user. For more about this, see *Section 15.7.1, "Assigning Object Rights to User/User Groups"*.
  - **Domain Rights...:** This page can be used to assign Tenant Domains rights to the user. For more about this, see *Section 15.8.5, "Assigning Domain Rights"*.

  Important Note:
  Both rights pages only display the rights that are *directly* granted for the user. Rights that are granted by the membership to groups are not displayed here.

14.1.1 Creating a New User

The **User** submap is used for all administration tasks concerning single users. It can be opened with a double-click on the User symbol in the User Administration submap.

The User submap has the menu item **New->User** in its context menu. This entry opens an input mask where the new user data can be entered (*Figure 20*) which is located on the tab **Information**.

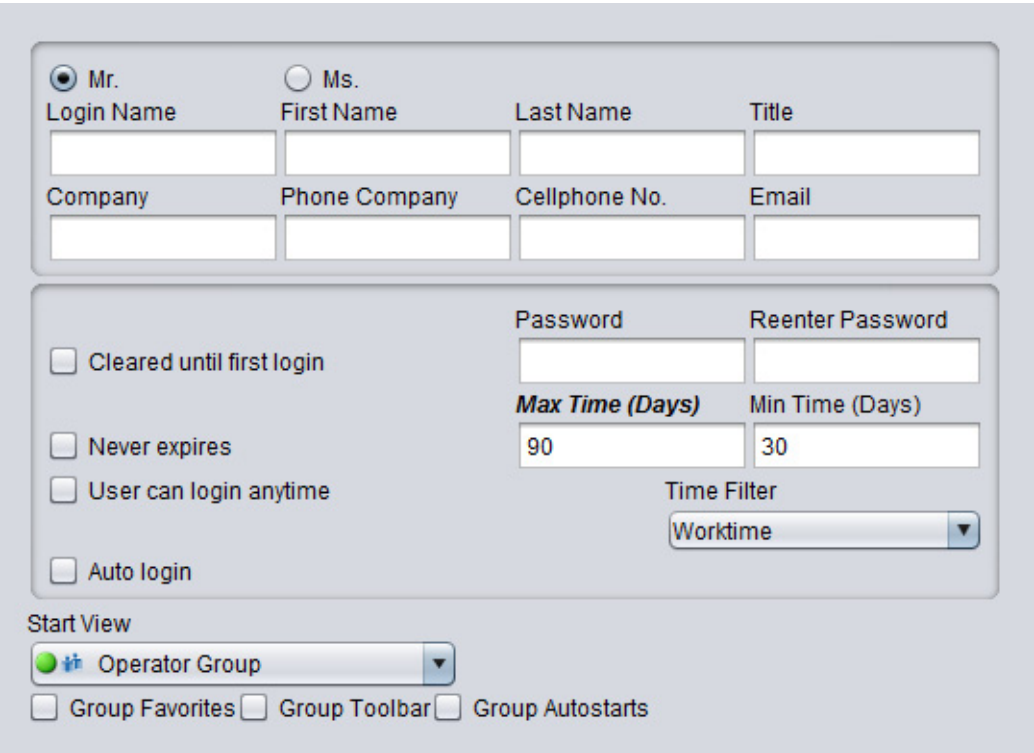


Figure 20 Addition of a new user

Fields:

- **Login Name**, **First Name**, **Last Name**, **Title:** enter the indicated parameters here.
- **Company**, **Phone Company**, **Cellphone No.**, **Email:** enter the indicated parameters here.
- **Password**, **Reenter Password:** enter password and confirm. To prevent the usage of insecure passwords, passwords have to follow a number of guidelines (see *Section 14.1.2, "Password Policies"*).
- **Max Time:** the maximum number of days the password will be valid; after that time the user will be forced to set a new password.
- **Min Time:** the minimum time the password must be valid; it will not be possible for the user himself to set a new password before the indicated period of time is over.

Check boxes:

- **Cleared until first login:** the password will be empty; when the user logs in for the first time, he/she will be forced to set a password.
- **Never Expires:** the password which has been set for the user will never expire; it can be changed whenever the user or an administrator desires to change it though.
- **User can log on any time:** the user may log on whenever she wants. If this box is not checked, time controlled login/logout will be enabled. More about this and the **Auto Login** can be found in the *Section 14.1.3, "Time Controlled Login/Logout"*.

Start View

The selection menu and the control fields located below it determine whether the settings for the favorites (see *Section 5.15*), the toolbar (see *Section 5.15.2*) and the user's autostart (see *Section 5.15.1*) are to be defined individually for the user, or whether the settings are to be determined via a selected group (see *Section 14.2*).

In the selection menu, a **Start View** can be assigned to the user. This can be the *Root* node of the navigation tree, the *User's* individual view, or any defined group.

If a group is selected, the **Group Favorites**, **Group Toolbar** and **Group Autostarts** checkboxes are active

If this is the case, the settings of the user will be used for unmarked fields and the settings of the selected group will be used for marked fields to define the user‘s Favorites, Toolbar and Autostart.

Groups

The page **Groups** that is only available during the creation of a new user can be used to specify to which groups the current user is assigned. For changes at a later time, the usual Group functions can be used (see *Section 14.2.2*).

The page contains a list of all defined groups, with corresponding control fields by which the desired group memberships are displayed or determined.

The entry **Group Assignment based on start view** can be selected to assign the user to the group that has been selected as the **Start View** on the **Information** page (see above).

14.1.2 Password Policies

To ensure the usage of ’secure’ passwords, passwords have to follow a number of policies. These are:

- A password must have a length of at least 8 characters.
- It may not have a length of more than 16 characters.
- It must contain at least one upper case letter (A-Z).
- It must contain at least one digit (0-9).
- It must contain at least one special character
  ’! $ % & ( ) = ? / * - + @ # < >’.
- It may not contain a sub string that consists of three or more identical characters (no character may be immediately repeated more than once).
- A new password must differ from the last 5 passwords defined for the user.

After five successive failed different login attempts, a user will be locked. The user can be unlocked by an Administrator to be able to login again. Else the login will be unlocked after 60 minutes or aften a server restart.

14.1.3 Time Controlled Login/Logout

The OpenScape FM Desktop provides an operation that time ranges are taken into account for the login/logout mechanism to the OpenScape FM server for a user. This operation is enabled by **not** checking the **User can log on any time** checkbox in *Figure 20*.

When this mechanism is selected a time filter can be chosen for the user by selecting it from the **Time Schedule** selector in the same window. This time filter contains time intervals and a time zone. The user will only be able to login during the intervals defined by the chosen time filter. How different time filters can be configured is described in *Chapter 20, “Time Schedule”*.

Whenever a time restricted user initiates a login to the OpenScape FM server, the server verifies based on the chosen time filter whether the current date lies within the allowed time intervals. If this is not the case, the login is rejected. Otherwise the login is granted and the OpenScape FM server evaluates the next logout time and provides it to the client. If this time is reached, the client initiates an auto logout from the server.

If the **Auto Login** checkbox is selected for the user, during auto logout the server will calculate the next time the user is allowed to login to the server. The OpenScape FM client will then wait the calculated time period and will then reconnect the user automatically to the server.

If a non time restricted user logs on to the server the auto login/logout mechanism will not be activated.

Important note:
While the OpenScape FM client is disconnected from the server and is waiting for an auto login, configuration changes for the time filter chosen for the corresponding user are not considered. Auto logout times are evaluated during the login phase.

Example:
User 'foo' is allowed to work with the OpenScape FM from 7 pm to 9 am every day. She starts the client on tuesday at 8 pm and logs on to the server. The client performs an auto logout on wednesday morning at 9 am and reconnects automatically (if enabled) at 7 pm in the evening and so on.

14.1.4 The User "root"

When you have installed OpenScape FM, the "root" account has been created automatically. In order to start work with OpenScape FM you have to log in as "root": leave the **Password** field empty and press **Login**. Then you will be asked to set the "root" password. Now you can go ahead and create further user accounts.

14.1.5 Assigning Rights to a New User

When a new user is created, he has only restricted rights. He has the right "Base->Map-Reader" for all objects, the right "User" for the root object and the right "Base->Change Password" for itself. To learn about how to assign more rights to the user please proceed to *Section 15.7.1, "Assigning Object Rights to User/User Groups"*.

14.1.6 Deleting a User

To delete a user the entry **Edit->Delete Object** can be used from the context menu of the user's symbol. All users besides "root" can be deleted.

14.1.7 Changing the Password

To change the password for a user, the symbol representing the user's account has to be located. From the symbol's context menu the entry **Password->Change** can be selected to enter a new password for the account (*Figure 21*).

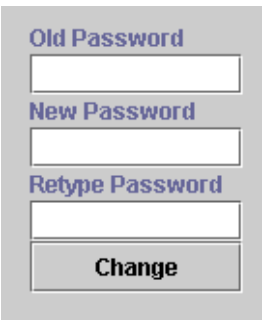


Figure 21 User Administration: new password

To prevent the usage of insecure passwords, passwords have to follow a number of guidelines (see *Section 14.1.2, "Password Policies"*). The changes will take effect on the next login. Unless you have administrator rights (e.g. are logged in as "root" or another user with administrator rights), you are only allowed to change your own password.

Another method to change the password of the currently logged in user is via the menu item **Client->Change Password** in the main menu bar.

14.1.8 Locking a User

With the context menu items **Lock User** and **Unlock User** a login lock can be assigned to the user, or the lock can be removed. If a user is locked the label of its symbol will be enclosed in asterisks and its status will be set to *Critical*. When the user gets unlocked the asterisks will be removed and the *Critical* status will be removed.

The user 'root' cannot be locked manually.

A user can lock himself by selecting the main menu entry **Client->Lock User**. After this, the user can only be unlocked by an Administrator.

Five consecutive login failures automatically locks a user to prevent further login trials. If the user ‚root' gets automatically locked, its account can be reactivated be restarting the OpenScape FM Server.

14.1.9 Authentication by Active Directory

In addition to internal user accounts that are only valid within the OpenScape FM, the Windows Active Directory Domain Service can be used to authenticate users.

To activate this function the Active Directory interface has to be configured and unlocked within the OpenScape FM. This can be done by selecting the main menu entry **Server->Administration->Server Properties**. On the page **Active Directory Configuration** the checkbox **Enable Active Directory Login** may be checked and the **Host**, the **Port** and the **LDAP Root Context** that should be used for the Active Directory Domain Service may be defined here.

The checkbox in front of **Enable Secure LDAP** can be used to determine whether LDAP over SSL (LDAPS) should be used (checkbox checked).

If no **LDAP Root Context** is defined, the whole Active Directory database is used to find the user. By entering strings like e.g. `DC=bui` the search can be restricted to a sub tree within the domain tree (in the example to domains starting with `bui`).

By removing the check mark **Enable Active Directory Login**, the Active Directory access can be locked at any time.

For the authentication by the Active Directory the login will be performed within the usual login window. In this case the Active Directory identification „`<DOMAIN>\<USER>`" will be used as the login name, and the respective Active Directory password has to be entered.

If an Active Directory user logs in to the OpenScape FM for the first time, a new Active Directory user account will be created within the OpenScape FM. This account will only allow Active Directory login in the future. The account, as any other account, can be locked or unlocked individually within the OpenScape FM by an administrator.

During each login to an Active Directory account, the groups currently assigned to the user within the Active Directory will be used for the current OpenScape FM session. In addition the user information will be synchronized and changed accordingly within the OpenScape FM if necessary.

Within the OpenScape FM, an Active Directory user gets an assignment of the groups that are assigned to him within the Active Directory at login. At least one of his Active Directory groups has to exist within the OpenScape FM or the login will not be possible and the user account will be removed within the OpenScape FM. The account will also be deleted when the respective user is removed from the Active Directory.

Important note:
The groups are not automatically created by the OpenScape FM. Groups with matching names have to be created manually and the corresponding rights have to be manually attached to the groups.

14.1.10 Exporting/Importing Users

The OpenScape FM provides a feature to export all defined users (with the exception of the user `root`).

The export is started by selecting the menu entry **Edit->Export Data...** within the context menu of the object **Root->User Administration->User**. If the export gets started, the name and location of the export file can be defined in a file browser. Subsequently the respective export file will be generated using the XML format.

The export file contains **all** users (except `root`), their configuration and rights settings. The passwords and password settings are **not** a part of the export.

By using the menu entry **Edit->Import Data...** within the context menu of the object **Root->User Administration->User** the user data can be loaded at a later time.

During the import, users for which an entry with the same name already exists, will **NOT** be overwritten. Instead their import data will be ignored.

Not existing users will be created according to the imported data. It has to be kept in mind that **NO** password will be assigned to these users. The newly created users should therefore be locked or a password should be assigned by the administrator.

If the exported user was only allowed to login during an active time schedule, the same will be true for the imported user. But only if a time schedule with the respective name exists during the import.

The export/import can e.g. be used when the rights for a user should be modified.

If the results of the modifications are not satisfactory, the respective user can simply be deleted. If the previously generated export gets imported, the settings of the deleted user will be recreated. The settings of all other users will not be touched by the import.

14.2 User Groups

A user group is a set of users. It provides a more efficient and flexible method to assign rights to sets of users. By assigning specific rights (object or tenant domain rights) to a user group all users belonging to this group will get this right granted (see *Section 14.2.4, “Assigning Access Rights to a User Group”*). When a new user is added to a group, all the rights currently assigned to this group will be automatically inherited by the user. The rights granted to a specific user will be the union of all the rights inherited from the user groups the user belongs to and of the user specific rights directly assigned to the user.

The “Groups” symbol is the starting point of the group administration. From here new user groups can be created and also users can be assigned to the created user groups. On its view all created user groups are shown.

In short the “Groups“ symbol offers the following operations:

New->Group...:
Via this menu item a new user group can be created.

Configure...:
This operation opens a dialogue where the relation between user and user group are shown and which allows to assign one or multiple users to one or multiple user groups.

14.2.1 Creating a User Group

In order to create a user group the menu item **New->Group...** from the “Groups“ symbols context menu has to be selected. A window will be opened where the name of the new user group has to be entered in the text field **Group Name**. Optionally a description about the user group can be entered in the text field.

After pressing the **Add** button the new user group will be added to the view of the “Groups“ symbol.

The **Cancel** button will close the window without creating a user group.

For a user group the following operations are available on different pages by using the menu entry **Configure...**:

- **Description**:
  Shows the user group name and the entered description.
- **Object Rights**:
  Via this menu item object rights can be assigned to this user group.
- **Domain Rights**:
  Via this menu item access rights for tenant domains can be assigned.

At the beginning, the newly created user group is „empty“, i.e. no users and access rights are member of it. The next step is to assign users (*Section 14.2.2, “Assigning Users to User Groups”*) and access rights (*Section 14.2.4, “Assigning Access Rights to a User Group”*) to the newly created user group.

14.2.2 Assigning Users to User Groups

The starting point for the assignment of user to user groups is the “Groups“ symbol on the “User Administration“ view. Its context menu offers the menu item **Configure...**. Selecting this menu item opens an info browser with a matrix showing which user is a member of which user group(s): each row represents a user, each column except the first represents a user group and each cell contains a checkbox which indicates whether a user is a member of a user group. If the checkbox is checked the user is a member of the user group otherwise not.

On the right hand side for each existing user group a checkbox and a button exist, these dialogue components are important for the assignment of user to user groups.

If no user groups exist only user will be listed in this info browser.

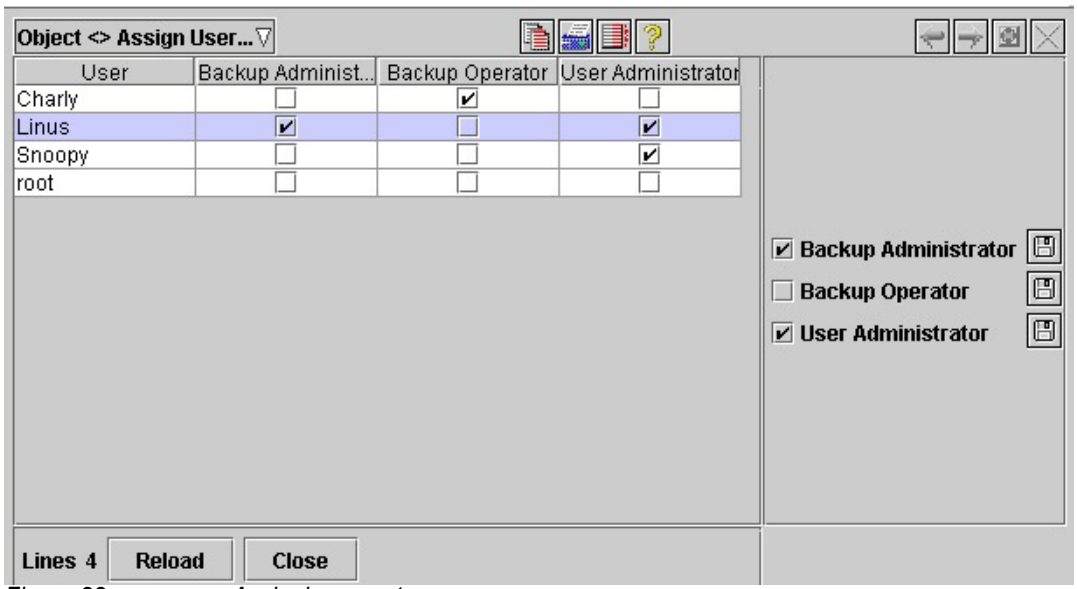


Figure 22 Assigning user to user groups

To assign a user to a user group select the row representing this user, check the checkbox of the user group on the right hand side of the dialogue and press the respective button.

Since this info browser supports multiple selection and each action done via the dialogue components on the right hand side effects all selected rows, multiple user can be assigned to one user group and even multiple users to multiple user groups.

Pressing the button will have the following effects:

- The matrix in the info browser will be reloaded.
- The respective user(s) will be added to the view of the “Assigned User“ on the related user group view. Therefore taking a look at the “Assigned Users” submap of a user group is a comfortable method to view which user are assigned to this group.
- On the view of the related user(s) the assigned user groups will be added to the container “Assigned Groups“. This container contains all user groups the user is a member of.

14.2.3 Remove a User from a User Group

In order to remove a user from a user group the following steps have to be performed:

- First you have to open the “Assign User“ dialogue, via the menu item **Configure...** from the „Groups“ symbols context menu.
- In the opened dialogue the row which represents the respective user has to be selected.
- Then uncheck the checkbox of the user group the user should get removed from on the right hand side, do not forget to press the appropriate  button thereafter, see *Figure 22*.

Now the user will be removed from the “Assigned User“ container of the user group, and the user group will be removed from the “Assigned Groups“ container of the user, too. As a consequence, all access rights which were assigned by the user group does no longer apply for this user.

14.2.4 Assigning Access Rights to a User Group

A user group by itself is not really useful. In combination with access rights it is a great assistance for the user administration. With user groups it is possible to assign access rights directly to a set of users instead of assigning the same rights to each user separately. And when a new user is added to a group it will automatically inherit all the rights currently assigned to this group additionally to the rights the user gets assigned directly or by other user groups.

The assignment of access rights to user groups works on the same principles as the assignment to single users, which means that the assignment of object and tenant domain rights is possible. The assignment will be described in detail in *Chapter 15, “Access Rights”*. Anticipatory it is to say, that for the assignment of access rights the starting point is the respective user group object and that a “Assigned Rights” object will be represented on the view of a user group object showing the rights granted directly.

14.2.5 Deleting a User Group

To delete a user group use the context menu of the user groups symbol and select **Edit->Delete Object**. The users which were members of this user group will be kept, but they will loose the rights which were assigned by this user group, i.e. they have then only the access rights that were assigned directly or by other user groups.

15 Access Rights

Within the OpenScape FM an **Access Right** provides the right for a specific user to perform a specific action on a specific object. It is therefore represented as a triple consisting of a **User**, a **Right** and an **Object**.

Virtually anything within the OpenScape FM is represented by *objects*. For example: objects represent physical devices that are monitored, containers to aggregate and structure objects, users, and even components of the OpenScape FM itself.

Rights exist for practically all possible actions that can be performed within the OpenScape FM. This can be the right to press a specific button, the right to activate a specific menu entry, or the right to see an object.

To perform a feature, the user needs to have the necessary rights for a feature *and* for the objects on which the feature should be used.

Example:
To add an ECE Module (see *Event Correlation Engine Plugin User Guide*) to an ECE Submap, Administrator Rights for the ECE Submap are needed. If the added module should monitor IP nodes, rights for the specific IP nodes are also needed.

Sets of Access Right Elements:

Since it would be a cumbersome to add or remove Access Rights for every specific user/right/object combination, place holders can be used for all three aspects of the triple to represent whole sets of users, rights or objects within an Access Right.

Groups (see *Section 14.2*) are sets of **Users**. When an Access Right is granted for a Group, it is granted for all members of the Group. A User may be a part of any number of Groups.

Domains (see *Section 15.8*) are sets of individual **Objects**. Access Rights granted for a Domain are granted for all objects within the Domain. An Object may be a part of any number of Domains.

A second method to define Object sets is the hierarchical assignment of Objects. An **Object Hierarchy** consists of an Object and all Objects that are located within the subtree of this Object.

Rights can be aggregated into **Roles**. A Role is a set of rights that should contain all rights that are needed to perform the tasks correlated with the Role.

Three basic Roles are predefined within the OpenScape FM:

- **Administrator**: Having this Role for an Object allows a User to perform all possible actions for this Object. This Role e.g. allows the configuration of SNMP parameters, the acknowledgement of events or the configuration of the network topology for the Object.
- **Operator**: This Role provides the rights to see the information, submap and events of an Object.
  This Role e.g. allows to see the configuration of the Object‘s SNMP parameters or the generation of a report that displays values of the Object.
- **User**: This Role provides basic access. It allows to see Objects but not to perform any actions on these Objects.
  This Role is needed to navigate through the navigation tree.

Instead to a User/Right/Object combination, an Access Right can e.g. be granted to a Group/Role/Domain combination. All Users within the Group will then have all Rights provided by the Role for all Objects within the Domain. This provides numerous User/Right/Object combinations with a single assignment.

The assignment of sets has an additional advantage: if Users are added to the Group or Objects are added to the Domain the same Access Rights will automatically apply for the new users or objects.

Revoking Rights:

Access Rights can be actively revoked for specific Users and Objects or Object Hierarchies.

Generally this is not necessary, since if no right is granted, this corresponds to revoking all rights.

But if e.g. a Right is granted for an Object Hierarchy, it may be desired to remove Access Rights for some members of the hierarchy.

Simplified Assignment:

While very complex Access Right structures can be generated within the OpenScape FM, mostly it is sufficient to use Object Hierarchies, Users or Groups and the predefined basic Roles. For this a simplified method to assign or revoke rights is provided (see *Section 15.1*).

Showing Active Rights:

For each Object the currently active Access Rights can be displayed (see *Section 15.9*).

Example:

An environment in which six team members should monitor a network is given. The network itself consists of two major regions: *Region West* and *Region East*.

Two members *Chris* and *Clair* should be in control of the whole network, its configuration and the administration of the OpenScape FM itself.

Two members *Will* and *Wilma* should monitor and handle *Region West*, but they should only see *Region East*.

Two members *Eric* and *Emily* should monitor and handle *Region East*, but they should only see *Region West*.

To realize this setting with minimal effort, the following steps can be performed:

1. Creation of Users: *Chris*, *Clair*, *Will*, *Wilma*, *Eric* and *Emily* (see *Section 14.1*).
2. Creation of Groups: *Control* with *Chris* and *Clair*, *West* with *Will* and *Wilma* and *East* with *Eric* and *Emily* (see *Section 14.2*).
3. Creation of a useful Network Topology (see *Chapter 16*): Creation of a container *Regions* that contains the containers *Region West* and *Region East* on its submap. The two sub containers should hold all objects that belong to the respective region.
4. Creation of seven Hierarchical Access Rights for the following triples (see *Section 15.1*):
   - Group *Control* / Role *Administrator* / Object Hierarchy starting with *Root*.
     This allows the members of Group *Control* to do virtually everything.
   - Group *West* / Role *User* / Object Hierarchy starting with *Root*.
     This allows the members of Group *West* to navigate the tree (should be set by default).

- Group *East* / Role *User* / Object Hierarchy starting with *Root*.
  This allows the members of Group *East* to navigate the tree (should be set by default).
- Group *West* / Role *Administrator* / Object Hierarchy starting with *Region West*.
  This allows the members of Group *West* to handle the objects in *Region West*.
- Group *West* / Role *Operator* / Object Hierarchy starting with *Region East*.
  This allows the members of Group *West* to see the object information within *Region East*.
- Group *East* / Role *Administrator* / Object Hierarchy starting with *Region East*.
  This allows the members of Group *East* to handle the objects in *Region East*.
- Group *East* / Role *Operator* / Object Hierarchy starting with *Region West*.
  This allows the members of Group *East* to see the object information within *Region West*.

To create these triples, the Hierarchical Rights for Standard Roles configuration has to be opened from the *Root* object for the first three triples, and from the *Region West* and *Region East* containers respectively for the last four triples.

15.1 Hierarchical Rights for Standard Roles

Often the general access rights for the roles *Administrator*, *Operator* or *User* should be granted or revoked for a complete subtree of objects.

In such cases hierarchical rights (see *Section 15.3*) for the basic roles can be directly assigned or revoked for a specific object and its child objects.

For this allocation the entry **Properties** has to be selected from the context menu of the specific object and the page **Object Rights** has to be selected within the property window.

The page displays one row for each known user and user group, showing the hierarchical configuration for the current object.

The column **Names** shows the names of the individual users (preceded by a pictogram of a single person) or user groups (pictogram of multiple persons).

The column **Permission** shows whether the user or user group currently has *Administrator*, *Operator* or *User* rights for the object, whether the access is *Denied* or whether no rule has been assigned (no entry).

When a right has been granted or revoked, the column **Inherited** shows the object for which the rule was defined. This may be either the object itself, or the object closest in the hierarchy above the object for which a hierarchical definition has been made for the specific user or user group.

The access rights for a user or user group can be changed by selecting the respective row and using the selection menu **Permission** and save button on the right side of the table.

When *Administrator*, *Operator* or *User* is selected and saved, the corresponding Access Rights for this role will be hierarchically assigned to the current object and for the selected user or user group. Therefore the entry in the column **Inherited** will change to the current object.

If the right is granted by being the member of a group, then a group symbol is displayed and the column contains the name of the group by which the right was assigned.

The selection *Deny* works accordingly, but hierarchically revokes all role rights.

The selection *Clear* removes the currently displayed setting. Even when the rule has been assigned for another object than the current. The rule that is now effective will be displayed instead.

Important Note:
Selecting *Clear* may have undesired side effects when the inheritance does not result from the current object. In this case, not only objects that are located below the current object are affected by the deletion. In the extreme the access rights for all objects might be affected.

15.2 The Rights Hierarchy

The OpenScape FM Desktop offers various access rights which affect the representation of the main menu bar, object context menus, the right to execute actions on objects, the visibility of objects on submaps and the visibility of events in the event browser. For example, if a user does not have all rights for the main menu bar he will see only a subset of the menus.

Generally access rights can be assigned to user and user groups. User groups are a set of user.

By assigning or not assigning specific rights for certain objects to a user/user group, an administrator can individually restrict the access of user to the OpenScape FM system. The assignment of a right is basically done by selecting an existing user or user group, opening the “Assign Object Rights” browser, selecting a specific right and a related object in that browser (see *Section 15.6, “Assigning Rights”*). In this way a specific access right relation for a certain user/user group, object and right is set up.

In case of user groups all user belonging to this group will get this right granted. When a new user is added to a group the user will automatically inherit all the rights currently assigned to this group. The rights granted to a specific user will be the union of all the rights inherited from the user groups the user belongs to and of the user specific rights that are directly assigned to the user.

The OpenScape FM Desktop provides six main right categories: “Administrator”, “Domain Administrator”, “Customer Administrator”, “Technical Administrator”, “Operator” and “User”. These main rights are an hierarchical aggregation of basic rights and define so called “roles“.

The following predefined administrator roles are to be used when a Secure Single Sign On (SSSO) access to specific element manager has to be configured:

Administrator, **Domain Administrator**, **Customer Administrator** and **Technical Administrator** are the super administrator which shall have read/write access to all OpenScape FM Management Systems, to the OpenScape FM Management applications, to the Element Managers and to the administrator profiles, including multi tenant configurations.

The role a user owns for a specific object will be passed during the invocation (SSSO) of the element manager which has to evaluate the role passed.

A user which has the **Administrator** role assigned for a certain object has full access to all functions provided by this object.

The **Operator** role enables a user to see the submaps and events of objects and to retrieve information from objects he has got this role assigned for. An Operator will not be allowed to perform actions and modifications on the objects. For example, for HiPath 4000 systems he can see the submaps of the HiPath 4000 systems, he will be able to retrieve fault, topology, hardware and software information from the HiPath 4000 system, but he will not be allowed to start discoveries, to change the placing in the topology tree or to perform unmanage/manage operations.

Another role is the **User**. This role can be considered a base type. The user can open all submaps of objects he has got the role **User** for. He will see all symbols (systems) on these submaps. However, he cannot perform actions on these symbols because he will not see the related context menu.

A user may have specific rights for distinct objects. He may act in the role “Administrator” for object “A” and in the role “Operator” for object “B”.

The rights are, like all objects in OpenScape FM, arranged in a hierarchical order. OpenScape FM is a sophisticated application which is based on the object oriented model. Therefore, the rights themselves form an object hierarchy. This Rights Hierarchy provides an overview over all basic and plugin-specific rights. Each module, like the Desktop and the various plugins, contain specific rights for viewing and changing access to the OpenScape FM system.

Of course, an administrator can also assign all the fine-tuned rights from the Rights Hierarchy.

15.3 Scopes for Rights

Rights are defined for individual functions or whole groups of functions that can be performed within the OpenScape FM. Each right is represented by a symbol within the rights navigation tree.

Access rights rules allow or deny the use of these functions for specific objects or object groups by specific users or user groups.

Each rule therefore consists of four base parameters:

- The right (the functionality that should be allowed or revoked).
- The user or group (who should or should not use the functionality).
- The object or objects (for which the function should be allowed or denied).
- The rule type (allowing or denying).

There are three possible options to assign objects to a specific rule:

- A rule can be assigned to *All* objects.
- A rule can be assigned to *One* specific object.
- A rule can be assigned to an object *Hierarchy*. That means to a specific object and all the objects within the object subtree below the object.

When a hierarchic assignment is used, the rule will be assigned to all objects within the respective subtree. The assignment will be removed for objects, that are moved out of the subtree, and it will be automatically added to objects that are moved to or created within the subtree at a later time.

Note:
Rules for user defined groups of objects (called Domains - see *Section 15.8*) are a special case of hierarchic rules. These rules will be hierarchically assigned to the respective domain object which contains all objects within the domain in its submap.

How rules are evaluated will be explained in the following *Section 15.4*.

15.4 Evaluation Order for Rights

This section describes how the defined rules for access rights are evaluated to decide whether a specific functionality might be performed by the current user of the OpenScape FM.

To perform a function on an object, two conditions have to be fulfilled:

- At least one rule exists that allows the desired function for the current object and the current user.
- No rule with equal or higher precedence exists that denies the function for the current object and the current user.

Generally more specific rules are of higher precedence then lesser specific rules. For example, rules for a user are more specific than rules for a group of which the user is a member. Rules for an individual object are more specific than hierarchical rules for an object container, which in turn are more specific than hierarchical rules for an enclosing object container.

Example:
In the case that the container *AllRegions* contains the containers *RegionOne* and *RegionTwo*, and a user should only see the objects of *RegionOne*, it is possible to assign an allowing hierarchic rule to the container object *AllRegions*, and a denying hierarchic rule to the container object *RegionTwo*.

The access rule evaluation performs the following steps in order to decide whether a function may be performed by the current user and for a specific object.

If a denying rule matches within an execution step, the access will be denied.

If an allowing rule and no denying rule matches within an execution step, the access will be allowed.

When no rule matches within an execution step, the next step will be performed.

If there are no further steps and no matching rule has been found, the access will be denied.

Execution steps:

1. Rules for the current user, the specific function and *All* objects are checked.
2. Rules for the current user, the specific function and the *One* specific object are checked.
3. Rules for the current user, the specific function and all minimal *Hierarchic* sets of objects that contain the specific object are checked. Since the object can be represented by various symbols, this might be more than one set.

   A set is minimal, if no other hierarchic set exists that contains the specific object and that is fully enclosed by the set.

4. Rules for the current user, the specific function and all smallest *Hierarchic* sets of objects that contain the specific object are checked.

   A set is smallest, if no other hierarchic set exists, that has not been checked yet and that is fully enclosed by the set.

   This step is repeated until all sets are checked.

5. Steps 1 to 4 are repeated, but the specific function is replaced by its father function (the more general function that contains the current function in its submap).

   This step is repeated until there is no father function of the last checked function.

6. Steps 1 to 5 are repeated, but the current user is replaced by a group to which the user belongs to.

   This step is repeated with the next matching group until there are no groups left. The groups are checked in order of group generation.

15.5 Functions of the Rights

The descriptions of the various rights are displayed with the help of tooltips on the respective right symbols. These will work for right symbols within the object tree or on submaps.

The tooltips contain the following information:

- The name of the plugin and the right.
- The object type (**Target Object Type**) for which the right will be used.
- The information whether it is a single right or an aggregation of rights (**Category**).
- The **Scope** in which the right will be used (e.g. a right can effect a menu entry or it can activate a function/ process).
- A **Description** of the effects of the right.

The names of the rights of the Desktop begin with the plugin designation *Base*.

The names of the rights for of the IP Manager plugin begin with the plugin designation *IP Manager*.

Rights which will be added by activating additional plugins, will also be integrated into the right tree. They will be described by respective tool tips.

15.6 Assigning Rights

A right can be assigned to a user and to a user group. If more than one user should get the same rights for the same set of objects a user group should be created and the specific user assigned to this user group. The advantage would be that the assignment of the access rights has to be done only for this user group, instead for each user separately. If only one single user should get rights for a specific set of objects you can either assign the access rights directly for the user or create a user group with only this user and assign the specific rights to

the user group. The second has the advantage, that if at a later time another user should get the same role as this user the new user has only to be assigned to the created user group and inherits automatically the rights assigned to the user group.

However the assignment procedure for both (user and user group) is identical. The difference only lies in the starting point.

The assignment of rights to a user has to be done by locating a specific user (symbol) on the submap “User Administration->User” view, by opening the context menu and by selecting the menu item **Configure**. The pages **Object Rights** or **Domain Rights** are used for the respective configuration.
Both pages only display the rights that are *directly* granted for the user. Rights that are solely granted by the membership to groups are not displayed here.

The assignment of rights to user groups has to be done by locating a specific user group (symbol) on the “User Administration->Groups” view, by opening the context menu and by selecting the menu item **Configure** and the pages **Object Rights** or **Domain Rights**.

For both the same dialogue will be opened which differs only in their scope.

On the page **Object Rights** you can assign and delete the rights for one specific or for all objects (see also *Section 15.7, “Object Rights”*).

By selecting the page **Domain Rights**, you can efficiently assign and delete rights for a predefined set of objects, a so called “Domain". Domains are sets of objects (for more information please refer to *Section 15.8, “Domains”*). Before you can assign domain rights, you have to create a Domain (see *Section 15.8.1, “Create a Domain”*) comprising all objects that shall to be part of the domain. A Domain contains always the same set of objects for all user. The advantage of using Domains is, that you can very efficiently assign specific rights to user/user group for access to *all* objects in the Domain. This relieves you from the burden to select every single object for each user/user group separately.

Thus you can assign rights in two ways: for *specific* objects (Object Rights) and/or for *sets* of objects (Domain Rights), the Domains. A user/user group can have certain access rights for one or more objects and/or for one or more Domains.

Important Note:
There is no difference if a right for an object has been granted as an object right on the page **Object Rights** or as a right for a Domain on the page **Domain Rights**. The rights of one user for one object are the sum of the rights which have been granted as single object rights and the rights which have been granted as a part of a Domain right. If a right for an object has been granted more than once by different methods, that does not make any difference. “The right has been given to the user“ is the only fact that counts. Redundancies in the right assignment are possible.

Important Note:
If you want to give rights only for a certain object below the root container you should give at least **User** rights for all the objects in the hierarchy path to this object. These objects are the object itself and all its parent objects up to the home object of the map. Otherwise the user can not navigate to this object.

15.6.1 List Right Conditions For One User/User Group

In the OpenScape FM object hierarchy, every right which has been *directly* assigned to a user is represented by a „right granted“ object on the view of the “Assigned Rights“ container located in the users view. For a user group all rights that were directly assigned will be represented by a rights granted object. They are located on the view of the “Assigned Rights“ container which can be found on the user groups view. The rights granted to a user via a user group will **not** be added to the users “Assigned Rights“ container. Thus, in the Navigation Tree, you can gain a quick overview over all rights which have been granted to a user directly and which belong to the user group the user belongs to.

Rights that are granted to objects will be displayed with a *green* symbol.

Rights that are explicitly revoked for objects (see *Section 15.7.1*) will be displayed with a *red* symbol.

Rights that are granted for at least one object and revoked for at least one other object will be displayed with a *yellow* symbol.

Each right’s context menu offers the menu item **List Right Conditions** which opens a list with all object attributes (names and domains) for which this right has been assigned to or revoked from this user/user group. The column **Value** specifies the name of the object.
The columns **Table** and **Column** are displayed in *green* when the right is granted for the specific object. They are displayed in *red* when the right is revoked.
The two columns contain the entries `*/*` when the right is assigned *globally*, the entries `group`/`leaf` when the right is assigned *hierarchically* or the entries `names`/`selectionName` when the right is assigned *individually* for the object.

15.6.2 List of User/User Groups For One Right

In order to see to which user/user groups a certain right has been given, select **List Users/Groups** in the context menu of the specific right: a list with all user and user groups which have this right assigned is displayed, whereby each right may be restricted to selected objects.

The column **Type** shows if the right has been assigned to a user (“User“) or to a user group (“Group“). The name of the user/user group will be listed in the column **Name**. When a selected right is assigned to all objects (by using the **>>>>** button in the “Assign Object Rights GUI“,) the checkbox in the column **Global** is checked. Otherwise the checkbox is not checked.

15.7 Object Rights

The following sections handle the object rights. They will describe how to assign (*Section 15.7.1, “Assigning Object Rights to User/User Groups”*) and how to delete (*Section 15.7.2, “Deleting Object Rights”*) them.

15.7.1 Assigning Object Rights to User/User Groups

The assignment of object rights to a user or user group starts from the submap of the user or group. First you have to locate the Users or Groups symbol in a view (i.e. in a tree or a submap) and open its context menu. Select **Configure** and the page **Object Rights** to open the Object Rights interface. Select the right you would like to assign to the user/user group in the **Rights** tree (left-hand side). Then select the object for which you want to assign this right in the **Objects** tree (center). These trees are special navigation trees which represent the hierarchical OpenScape FM system of rights and objects. An Assignment of a certain right also includes the child rights.

It has to be kept in mind that the same does **not** apply to the **Objects** tree: within the **Objects** tree the child objects are **not added automatically**. You have to select every object, for which the right should be granted explicitly. Multiple selection is possible with the `ctrl` key resp. the `shift` key, so you can assign one right to one user for several objects at a time.

When you have selected the right and the object, click on the arrow button **Assign** / **>>** (which points to the right): the object has been added to the **Objects For Right** column (right-hand side).

The selected objects for the marked right appear in the list, **Objects For Right**, on the right-hand side.

In order to assign a right to a user or user group globally (for all objects), press the button **>>>>** (assign global). If there have been other objects those will be colored grey. Anyway for those objects the right is also assigned. They will be hold in the list, because when you remove the global assignment with the **<<<<** button (remove global), you might want to re-use your former object selection.

Important Note:
If a right should be granted for an object below the root container, at least the **User** right should be granted for all objects that are hierarchically between the object and the Root container. This means the object itself and all parent objects up to the root of the map. If this is not done, the user cannot navigate to the specific object.

Example:
If a user should get **Administrator** right for a network, he should get **User** right for the root object and the network topology.

Revoke Rights:
By using the buttons **!>** or **!>>>** a right for a single object or a object tree can be revoked. This makes it possible to negate a right that has been granted globally or by a parent object for specific objects. To define whether a right has been granted for an object or not, always the more specific rule is used. This means, the path will be followed up from the object to the root, until an including or excluding right assignment is found.

Display Rights:
You can see the rights of each user or user group by opening the view of the user or user group you are looking for: every right you have granted is displayed as a granted right icon in the “Assigned Rights“ container, whereby each granted right may be restricted to certain objects.

15.7.2 Deleting Object Rights

In order to remove a right or a right revocation for a certain user or user group, open the Object Rights interface via the users or user groups context menu. Select the right in the **Rights** column. Now the objects for which the right is granted are listed under the **Objects for Rights** column. Mark the object(s) on which the right should no more affect and press the arrow button **Remove** / **<<** , which points to the left,.

If you have assigned the right for all objects (over the **>>>>** Button/ assign global) you have to click the **<<<<** Button (remove global), to remove the right for them.

A second possibility to delete a granted right is via the standard menu item **Edit->Delete Object**, here the right is deleted for all objects for which is was granted (for the related user).

15.8 Domains

Some organizations operate large networks with a staff of service technicians. These organizations may have the requirement that certain groups of technicians are allowed to perform maintenance operations on specific subsets of the whole network. Additionally, not every technician in a service group shall be allowed to perform every maintenance operation. Hence, the assignment of specific rights to different persons for the same subset of the network is a mandatory requirement.

In such an environment multi domain capability is the solution: it enables to define sets of objects (called Domain) and to assign user/user group specific rights for all objects in a Domain. That way, an administrator or service technician can for example manage the system of a certain network “A“ (contained in Domain “A“) but he/she is not allowed to access data of any other network “B“ (contained in Domain “B“).

In OpenScape FM, this is achieved by forming so called “Domains”. A Domain is a set of objects.

Important Note:
Considering the rights of a user/user group for an object, it does not matter if she/it has received this specific object rights as dedicated object rights or he/it has been assigned the right for the Domain the object is part of. The user rights for an object is the sum (the set union) of all rights which have been granted either as single rights (via the page **Object Rights**) or as Domain rights (via the page **Domain Rights**).

In order to grant rights for a Domain to a user/user group, you have to perform two steps: create a Domain (see *Section 15.8.1, “Create a Domain”*) and assign rights for this domain to the user/user group (see *Section 15.8.5, “Assigning Domain Rights”*). Changes in Domains will immediately be propagated to the granted Domain rights of user/user groups.

Via a main menu option you can get a quick overview over all existing Domains (see *Section 15.8.4, “Overview Over All Domains”*).

15.8.1 Create a Domain

The central object for domains is located within the navigation tree at the position **Root->User Administration->Domains**.

New domains can be created by using the entry **New->Domain** from the context menu of this object. This will open a window in which a unique name and a description can be assigned for the new domain.

A new Domain container object will be placed on the submap of the Domains object.

15.8.2 Add an Object to an Existing Domain

All objects that belong to a domain have a copy of themselves on the submap of the respective domain container object.

Objects can be added to a domain by using Copy/Paste or Drag&Drop within the navigation tree to add them to the submap of the respective Domain symbol.

Alternatively the entry **Properties...** can be used from the context menu of an object that should be added.

The page **Domains** within the opened window displays a list of all defined domains. The button **>>** can be used to add selected domains to the current object. Double-clicking an entry moves it from one list to the other.

15.8.3 Delete an Object From a Domain

Objects are removed from a domain by removing their symbol from the respective domain container object. This can be done by selecting the menu entry **Edit->Remove Symbol**.

Alternatively the entry **Properties...** can be used from the context menu of an object.

The page **Domains** within the opened window displays a list of all defined domains. The button **<<** can be used to remove selected domains from the current object. Double-clicking an entry moves it from one list to the other.

15.8.4 Overview Over All Domains

The main menu item **Server->Administration->User Administration** provides a navigation tree with container **Domains** which holds all Domains defined and the contained objects.

15.8.5 Assigning Domain Rights

The process of assigning Domain rights to a user/user group is very similar to the process of assigning objects rights (*Section 15.7.1, “Assigning Object Rights to User/User Groups”*). First you have to locate the symbol of the user/user group you want to give access rights to and choose **Configure...** from the context menu and switch to the page **Domain Rights...**.

Rights can be granted or revoked using the same functions as described for object rights. In this case, in the tree **Domains** the domains are selected, for which a right should be granted or revoked.

The rights will then be granted or revoked for all objects that are part of the respective domain at the moment when the right is needed.

15.9 Current Access Rights

A number of rules might effect the access rights for a single object and its functions.

Rules applying to the object might be defined for the object itself, hierarchically or globally.

Different users or user groups may have different role rights for the object or even access denials.

To get an overview about the access rights that are assigned for a specific object, the entry **Properties** can be selected from the context menu of the object and the page **Assigned Rights** has to be opened.

The page displays a table with one row for each access rights rule pertaining to the current object (see *Section 15.3*):

The column **Type** displays whether the rule has been assigned specifically for the object (*Object*), hierarchically (*Hierarchic*) or whether it was assigned to all objects (*Global*).

The column **Name** displays the name of the user or user group for which the rule has been assigned.

The column **Right** shows the name of the access right or role that is granted or revoked by the rule.

The column **Object** lists the object for which the rule was assigned. In the case of hierarchical assignments this is the root object of the hierarchy. The entry is green for allowing and red for denying rules.

Important Note:
The page does not display the effective rights, but all rules that might affect the current object. The effective rights can be derived from this list (see *Section 15.4*).

Important Note:
The page may contain revoking rules that were not explicitly defined, but were implicitly generated by the system. E.g. the role *Operator* generated a denying rule for the role *Administrator* for the same object.

16 Network Topology Management

The Topology Manager is used by the OpenScape FM Desktop plugins as the common basis for a unified topology representation. The Topology Manager will initially create a new object on the root view which is labeled ‚*Network Topology*‘. This object and the object tree below it is the central repository for all topology objects.

The Topology Manager is responsible for the representation and management of complex hierarchical topologies consisting of nodes, edges and topology container objects. Nodes and edges are created by the OpenScape FM Desktop technology plugins. Nodes represent real network elements and their subcomponents like subsystems, boards and software modules. Edges represent physical and logical (communication) links between nodes. E.g. for HiPath 4000 systems an edge is used to represent a connection between a source and a target HiPath 4000 system (node). The Topology Manager evaluates edges to build a hierarchical structured connectivity representation.

In the course of structuring nodes and edges into a hierarchy container objects are created implicitly. They are managed by the Topology Manager. This hierarchy is freely configurable and can reflect e.g. geographical, administrative or other organizational structuring principles. There are two types of container objects: network and subnetwork container. The purpose of these two container types is to allow an optimized representation of a deep network hierarchy.

Network and Subnetwork Ids provide the means to define a path of a hierarchical tree. Based on the defined Network Id - and Subnetwork Id - assigned to a node, the Topology Manager will automatically create the related network - and subnetwork - containers in the network tree and assign the corresponding node to the last container of the path. Each node has one parent, e.g. its (sub)network container and can have children e.g. subcomponents.

A new Topology Container can be created by using the menu item **New->Topology Container...**. New nodes can be added to the container by using the menu entry **New->IP Node...**. Already discovered nodes can be added by using Copy/Paste functions or by dragging the items into the container‘s submap.

16.1 Topology and Hierarchical Network Structure

Nodes, edges and container objects are the basic building blocks of a hierarchical network structure. A container represents either a network or a subnetwork and is managed by the Topology Manager. Nodes can represent IP nodes or other systems (e.g. HiPath 3000 systems, HiPath 4000 systems). Nodes are managed by their related plugins, e.g. the IP Manager plugin is responsible for the management of IP nodes.

A Network Id and/or a Subnetwork Id can be assigned to a node. This assignment positions the node in the topology tree and the Topology Manager takes over the responsibility for the management of this positioning. The assignment of a Network Id (and Subnetwork Id) can be effected by a the IP-Discovery rules or by the user (see IP Manager Plugin User Guide). If no Network or Subnetwork Id is assigned to a node, it is not managed by the Topology Manager, and it is only visible on the plugin specific views (e.g. an IP node is part of its parent IP node container).

Right from the start most IP nodes are not managed by the Topology Manager. When they are discovered they are first added to the IP container view (an IP container is the child of an IP network). Only if IP discovery rules, which define the assignment of a default Network/Subnetwork Id, are found, they will be managed by the Topology Manager.

There are certain network elements which cannot be freely positioned in a topology tree. E.g. subcomponents of a node have a fixed relation to their parent node, their positioning can not be changed by setting the Network and Subnetwork Id.
IP networks are created by the IP Manager and are treated like nodes. An IP network can also be positioned in a topology tree. However, the related IP container has a fixed relation to its IP network, it cannot be positioned separately.

16.1.1 Configuration of Hierarchical Networks

As mentioned OpenScape FM Desktop supports a hierarchical structuring of networks to enable administrators to build a clearly arranged graphical representation. This is especially very useful when large networks have to be graphically represented in a comprehensive way and different views of abstractions are used by the network administrators. Enterprise specific structuring principles can be chosen: e.g. structuring of a company network on a geographical basis or a structuring reflecting the organizational units of an enterprise. In the first case each country could be represented by a topology container, in the latter case each department could be mapped to its own topology container. The next sections offer further information about this subject.

A network is structured by positioning the nodes (network elements) of the network within a conceptual hierarchy defined by an network administrator. There are two generally used parameters which allow to specify the position of a node in a hierarchy: the Network Id and the Subnetwork Id. For the sake of completeness the Primary Domain Id is mentioned here too, although it is not mandatory for the hierarchical structuring.

- The **Network Id** defines a path (see below) through a hierarchy/tree to a topology container in which the node shall be placed.
- The **Subnetwork Id** provides an additional means for structuring network hierarchy. The main advantage of using a subnetwork is to shorten the label describing the path (see *Figure 23*). Further, subnetworks are represented by a specific subnetwork symbol.
- The **Primary Domain Id** is used with edges and identifies the Target Domain in which the target of the edge is to be found. In *Section 16.4, “Domain Ids and Target Domain Ids”* this is described in more detail.

Network Ids and **Subnetwork Id**s define a path through a tree. A path can consist of a series of path names which are separated by the character “/”. The first path name defines the root, the last the target topology container. Each path name component identifies a container object which is graphically represented by a network symbol. All nodes with the same Network Id (and Subnetwork Id combination) are grouped together in the same target topology container and will be shown on the submap of this container.

In the example shown in *Figure 23*, for the node ’System 121’ one network and two subnetworks would be created: a network “Country A” with the child object /subnetwork “State P” which in turn has a child object/subnetwork “State P/Town Y”. The object label already indicates the network hierarchy, which makes it easier to get a picture of the network structure of large systems at a glance.

In order to use the "/" in the name, without creating a new topology container, you have to quote it with another "/".

One or multiple nodes can be assigned in one configuration step to a network hierarchy and topology container respectively. The next sections describe this subject in more detail.

When a new (sub)network hierarchy is configured by assigning nodes to a topology container or an existing topology container becomes empty, the topology will be updated right away.

16.1.1.1 Bulk Operation for Assigning the Network and Subnetwork Id

The topology network symbol's context menu is the starting point for all bulk operations. To change or look up topology parameters for multiple nodes of a topology container in one step, choose the menu entry **Properties...**. A browser/dialogue showing a list of topology parameters of all nodes of the container is displayed on the page **Topology->Network Configuration**. You can then go ahead and select one or more nodes from this list and set a new Network Id and/or Subnetwork Id. Press the button to complete the operation. The Topology view is updated immediately.

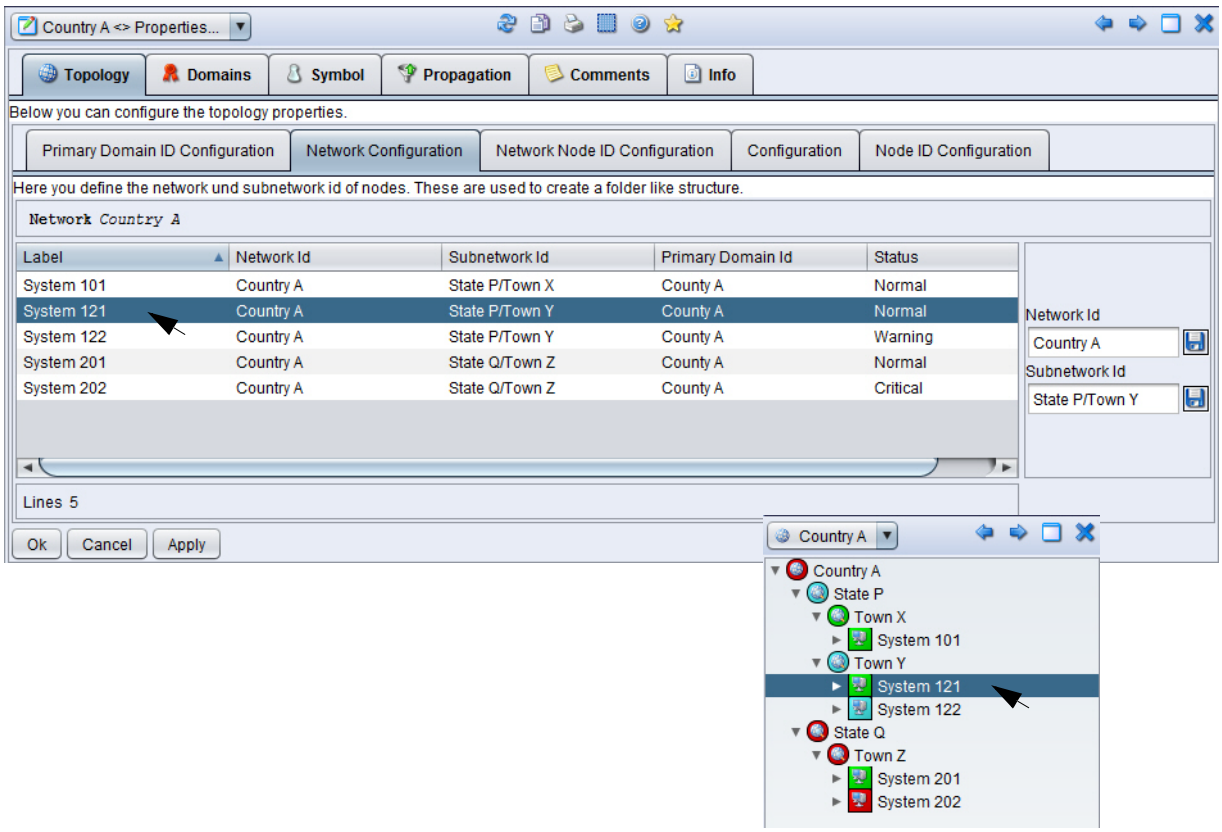


Figure 23 Configuration of network parameters as bulk operation

16.1.1.2 Bulk Operation for Assigning the Primary Domain Id

In order to change/set the Primary Domain Id for multiple nodes of a topology container choose the menu item **Properties...** from the network containers context menu. A browser opens that lists parameters for all nodes contained in this network on the page **Topology->Primary Domain ID Configuration**. To set the Primary Domain Id for specific nodes, select the corresponding lines, enter the Primary Domain Id and press the button.

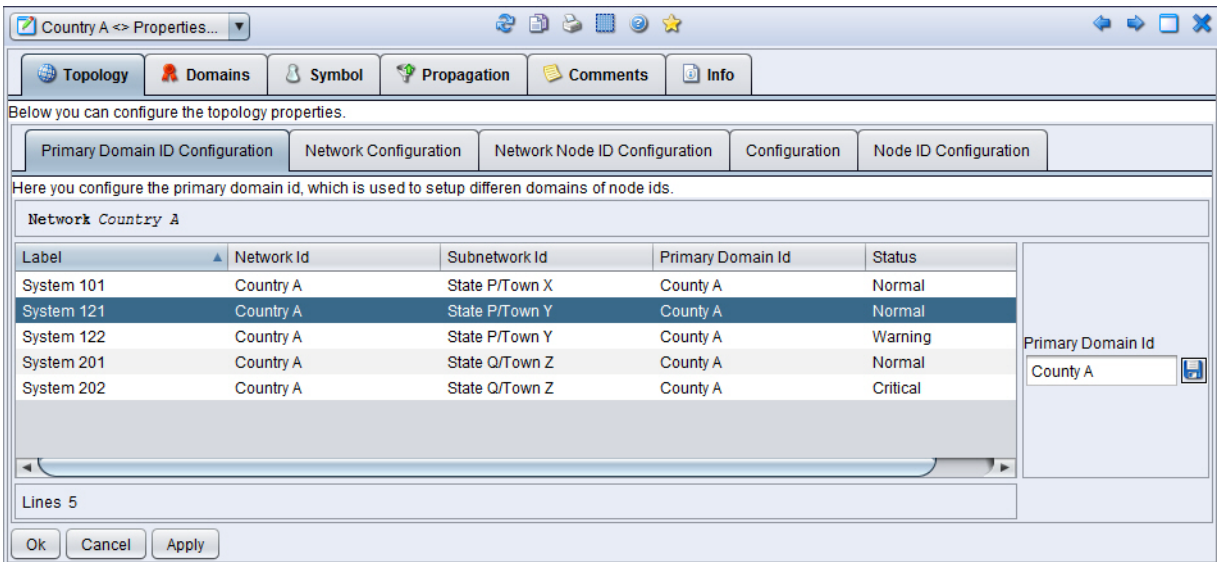


Figure 24 Bulk operation for Configuration of Primary Domain Ids

16.1.1.3 Assigning Network/Subnetwork and Primary Domain Id on Node Level

Of course, you can define the Network/Subnetwork and the Primary Domain Id in the context of a specific node. Choose **Properties...** from the node's context menu and enter the parameters for Network, Subnetwork, Primary Domain ID on the page **Topology->Configuration**.

If several nodes are selected, this configuration can also be performed using the menu item **Properties** of one of the nodes. Individual nodes can be selected within the list displayed on the page **Topology** and the individual values can be adjusted using the fields located beside the list.

16.2 Topology Edges and Hierarchical Connectivity

A topology edge is used to represent a connection between a source node and one or multiple target nodes. In this case the edge represents a directed connection, it starts at the source node and ends in the target node. A directed edge represents a specific connectivity type which is bound to the specific source node. It connects the source node to its target nodes but not vice versa. Directed edges are for example used to represent the connectivity of HiPath 3000 ports or HiPath 4000 trunk groups.

When the target node has also a directed connection to the source node, the edge is bi-directed. IP Interfaces are an example for bi-directed edges.

Edges get their source node assignment at creation time and this assignment will not change during its life time. The target node(s) of an edge object can change during its lifetime.

The target Node Id(s) assigned to an edge are used to determine the target node (or nodes) of the edge. The Topology Manager will automatically check if there are nodes in the topology having a **Node Id** which matches the edge's target **Node Id**. Then it verifies if the node is also in the same **Domain** as the edge (see *Section 16.4,*

“Domain Ids and Target Domain Ids”). If so, these nodes will be treated as the target node(s) of the edge. This is called the "implicit target node assignment".
There is also an "explicit target node assignment" which can be effected for example by a plug in (e.g. for IP interface edges) or manually by a user. An explicit target node assignment made by a user can be undone later on. If an explicit target node assignment is undone, existing implicit assignments will be reused by the topology manager to address the target node(s) of an edge. The configuration of an explicit assignment of a target node is explained in *Section 16.3, “Manual Explicit Target Node Assignment”*.

Typically an edge will address only one target node. But in some network environments multiple target nodes can be addressed by an edge. How multiple target nodes are handled and represented is described in *Section 16.5, “Handling of Multiple Target Nodes”*.

If no target node will be found for an edge, an ’External System’ object will be created. This ’External System’ object will be represented on the same view as the source node of the edge.

16.2.1 Meta Edges

Meta edges are used to represent the connectivity between two nodes in a topology hierarchy. A meta edge is an edge container which represents and contains one or multiple simple edge objects. If the last edge object is removed from a meta edge, the meta edge will be deleted.

Meta edges are only created by the Topology Manager to represent the hierarchical connectivity between two topology nodes/containers. Each meta edge has one source and one target node/container. Each edge which is managed by the Topology Manager, will be a child of at least one meta edge. Depending on the hierarchical network structure, it can be contained in several meta edges. The child objects of the edge are the source node and the target node defined by the target Node Id of the edge.

To describe why and when meta edges are created, the term “node sub hierarchy" is introduced. A "node sub hierarchy" describes a set of nodes. A "node sub hierarchy" related with a specific node contains this node as well as all nodes which are reachable from this node following the child relations and the "root node" (see below).

In *Figure 25* the node sub hierarchy of the subnetwork "State P" is shown. Members of this sub hierarchy are the subnetwork "State P" as the root node, all nodes contained in "State P", all lower-level topology containers and all nodes contained in these containers.

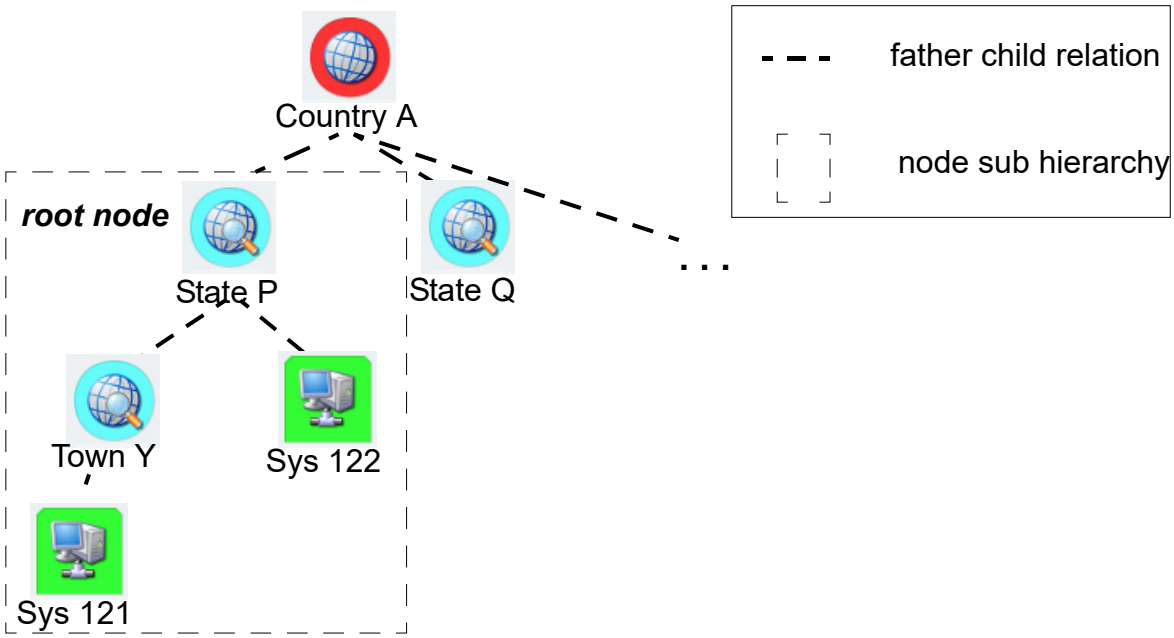


Figure 25 Node Sub hierarchy

A meta edge - connecting a node "Source" with a node "Target" - represents a connectivity between two "disjunct" node sub hierarchies. Disjunct means no node exists which is a member of both node sub hierarchies. The root nodes of a pair of connected node sub hierarchies have the same parent node. The root node of the node sub hierarchy containing “Source” will be called “source root” and the root node for “Target” “target root”.

The following figures present examples for different types of node sub hierarchies.

In *Figure 26* an example is shown where the source and the target node reside in the same subnetwork (view). They have the same parent node. Therefore, the node "Source" is at the same time the source node as well as the source root and the node "Target" is the target node as well as the target root.

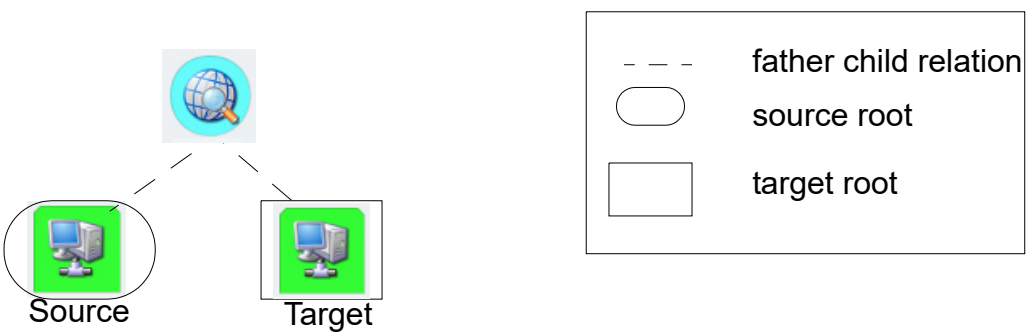


Figure 26 Target and Source node on same view

Figure 27 shows the first of three examples where the source and target node reside on different views. In this example the node Source belongs to the subnetwork "Town Y". Because Target and Town Y reside on the same view, they have the same parent, in this case the subnetwork State P. Therefore, the subnetwork Town Y is source root and Target is target node and target root.

Figure 27 Target and Source node on different views (1)

In *Figure 28* the node Source is both source node and source root and the node Target belongs to a subnetwork ("Town X") which is its target root.

Figure 28 Target and Source node on different views (2)

Figure 29 shows an example where "Source" and "Target" reside on distinct views, the subnetwork Town Y and Town X. As both subnetworks have "State P" as their parent node, "Town Y" is the source root of "Source" and "Town X" the target root of "Target".

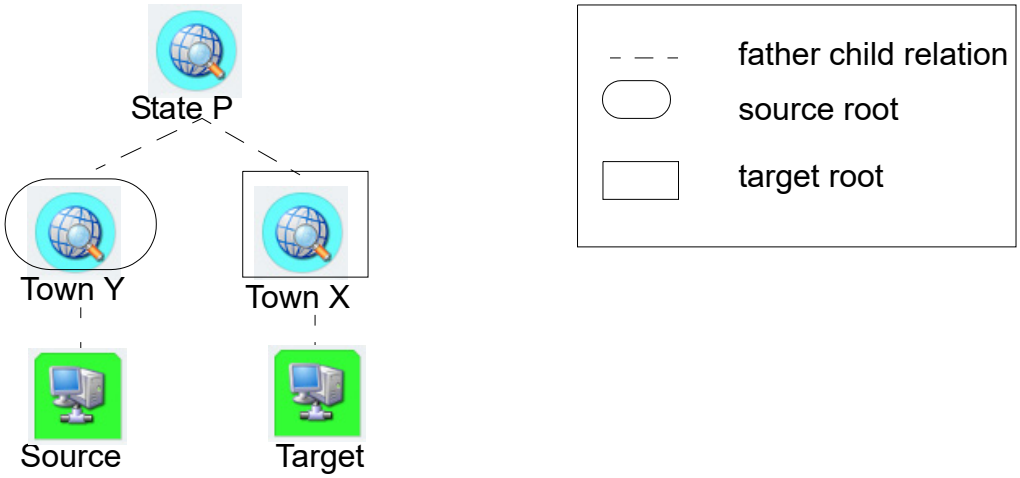


Figure 29 Target and Source node on different views (3)

When a source node is connected with a target node, meta edges are created from the source node, the source root and all topology containers in between to the target root. The target root is the target object of a meta edge and the source root determines how many meta edges will be drawn.The connectivity between the source node and the target node will be visible on all levels between the source node and the source root in the node sub hierarchy the source node belongs to.

For example if the node “Source” is also the source root, only one meta edge connecting “Source” and the target root of "Target" is created.

If “Source” is not the source root multiple meta edges will be created connecting the target root with

- the node "Source",
- the source root of the node "Source",
- and each topology container on the path from "Source" to the source root.

If the source object of a meta edge is not on the same view as the related target root, a **reference symbol** representing the target root is created and a connection will be drawn to this reference symbol.

16.2.2 Reference Symbols

Reference Symbols are place holders for symbols located on other submaps. They can be distinguished from normal symbols by a lighter color for their status. In addition their original label is enclosed in << >> brackets.

16.3 Manual Explicit Target Node Assignment

In some cases the implicit assignment of a target node which is done by the plugins can be overwritten by an explicit assignment of a target. In other words an existing edge should get another target node which has the effect that the former connection will be redirected to the new target node. For example a connection between the nodes

system1 and system2 can be configured to connect system1 with a system3. For the explicit reconfiguration of connections the following mechanism can be used. If it is feasible to assign a new target node for an edge on the physical level (e.g. direct configuration of a HiPath 4000 trunk) using configuration functions of the related element manager, it is recommended to use the element manager mechanism. We recommend the use of this method in the case that a configuration of the physical pendant of an edge is not possible, or systems of distinct technology types have to be connected.

Mark the desired target system via its context menu with **Edit->Link**. Then locate the edge for which the target node has to be modified and select **Edit->Set Target**. The target node of the edge is adjusted (to the selected system) right away and the topology representation is modified on the corresponding views (i.e. submaps and/or trees).

If the source and the target symbol are located on the same submap, the connection will be displayed between the symbols.

If the source and the target symbol are located on different submaps, the user can select, whether reference symbols for the target symbol and/or the source symbol should be added to the respective submaps.

In order to reset the connection select the edge again and select **Edit->Delete Object** from its context menu to remove the edge.

Important Note:
The usage of this mechanism only effects the OpenScape FM Desktop database. In order to change the target node for an edge permanently, functions of the technology type specific element manager have to be used.

16.4 Domain Ids and Target Domain Ids

In the following the usage of "Domain Id" and "Target Domain Id" will be motivated.

Domain Ids shall be used when there are identical node ids which has the effect that there is an ambiguous Node Id name space. Domain Ids are used in such a case to create unambiguous Node Id name spaces. A Node Id of a node is extended by a Domain Id. By using different Domain Ids domains with unambiguous Node Id name spaces are created. A node can be part of multiple domains.

16.4.1 Node Identification by Domain Id/Node Id

An edge connects nodes, a source node with a target node(s). The edge has a target Node Id and a Target Domain Id which are used by the Topology Manager to find the related target node(s) and to draw the connection(s) between the nodes. Edges representing a HiPath 3000 port or a HiPath 4000 trunk group have a related target Node Id. The target Domain Id of an edge is the Primary Domain Id of the related source node. In the simple case where all node ids are unambiguous and all nodes belong to one domain, the Topology Manager can assign each edge to the right target node. In network environments where identical node ids exist - multiple nodes having the same Node Id - an administrator has to ensure the correct assignment of an edge to one - or in the more complex case - multiple target nodes. The administrator has to qualify all nodes which shall act as a target of the edge. These are the nodes which have a Node Id equal to the target Node Id of the edge and the Domain Id of the related

Node Id has to be equal to the target Domain Id of the edge. Now the Topology Manager is able to create a relationship between the target Node Id and the target Domain Id of an edge with all nodes having a matching Node Id and Domain Id value pair.

A node has a set of "Domain Id/Node Id" pairs assigned. A "Domain Id/Node Id" pair can be assigned by a plugin or manually by a user. A Node Id can only be assigned in combination with a Domain Id. Normally all "Domain Id/Node Id" pairs of a node should be bound to the Primary Domain Id of the node, but there are some cases where this may not be the case, e.g. to represent the connections between the nodes of two separate domains.

All nodes which have a "Domain Id/Node Id" assigned which matches the "target Domain Id/target Node Id" of an edge are target nodes of this edge. In most cases only a single target node will be found.

Automatic Updates In Case of Changes to the Assigned Domain Id/Node Id Pairs:

If a new "Domain Id/Node Id" pair is assigned to a node, all edges which match to this "Domain Id/Node Id" pair will be drawn to this node. If one of these edges was connected to an "External System", the edge will get redirected to the new target node and the "External System" object will be removed.

If a "Domain Id/Node Id" pair is removed from a node, the target nodes of all incoming edges will be updated. If this node was the only target node of an edge, an “External System” object will be created now, to which these edge will be drawn. The "External System" object is labeled with the target Node Id.

16.5 Handling of Multiple Target Nodes

An edge which identifies its targets by a Node Id (e.g. HiPath 3000, HiPath 4000) can have multiple target nodes. If multiple target nodes are addressed by the target "Domain Id/Node Id" of an edge object, connections will be drawn to all target nodes. The label of the edge symbol shows the number of the target nodes. All source and all targets of an edge object will be represented on the view (e.g. submap) of the edge object.

16.6 Manual Indirect Target Node Assignment

A representation of a connection between systems of different technology types can be established by the manual assignment of a Domain Id/Node Id pair to the particular target node. This Domain Id/Node Id pair assignment will only be stored in the OpenScape FM Desktop database.

Important Note:
For system of the same technology type the most recommended method is to configure the target system directly at the edge, via its management tools.

16.6.1 Manual Node Id and Domain Id Configuration of a Node

The entry **Properties** of the node objects managed by the Topology Manager offers the page **Topology->Node ID Configuration**. This shows the following information:

Domain Id: Here the Domain Id of the single Node Id(s) will be listed.

Node Id: The Node Id will be found here.

Entry Type: Here it will be shown if the Node Id is set manually via a user or automatically via the plugin. When the Node Id is set automatically here the entry type of the Node Id will be found, for example for IP nodes the entry id is "ip internet". When the Node Id is created manually the entry type is "Manual".

Domain ID Writable: If the Domain Id is writable by the user this checkbox will be checked. For example the node ids of type "ip internet" are not writable.

Node ID Writable: If the Node Id is writable by the user the checkbox will be checked. Only manual created node ids are changeable.

Deletable: The Node Id can be deleted by the user when the checkbox is checked. Only manually created node ids can be deletable.

In a panel on the right side the **Domain Id** and the **Node Id** values can be changed. By pressing the **Create** button a new Domain Id/Node Id pair with default values will be created.

In order to change the values for an entry, select the row, enter the wanted values and press the  button next to the entered value.

If a manual configured Node Id is no longer required it can be deleted by selecting its entry and pressing the **Delete** button.

16.6.2 Copy and Paste the Node Id/Domain Id pair of an External System Object

Specifically for External System objects the Node Id/Domain Id configuration offers the possibility to copy the related Node Id/Domain Id pair and to assign it to another node. As a result the connection will be redirected to this node and the External System object will be deleted. The following actions have to be performed: The menu item **Topology->Copy Domain ID & Node ID** from the external (target) nodes context menu has to be selected (the Node Id/Domain Id of the external system will be fetched). Then the new target node - to which the information shall be assigned to - has to be located and its context menu item **Topology->Paste Domain ID & Node ID** has to be selected. The connection/edge will be redirected and in the “Node ID Configuration...” page of the new target a new entry reflecting this assignment will be found.

16.6.3 Manual Node and Domain Id Configuration for a Network

The Domain and Node Ids of nodes contained in a network can be changed via the network. In order to do this select the menu item **Configure** from the networks context menu and use the page **Topology->Network Node ID Configuration...** . The same browser as described in *Section 16.6.1, “Manual Node Id and Domain Id Configuration of a Node”* will be opened, but here all nodes of this network managed by the Topology Manager are listed. Select the nodes for which you want to change the values, enter the data and press the  button. The browser will be actualized and the topology will be reorganized according to the new values.

17 Help

The OpenScape FM Desktop offers three different help systems (see *Figure 30*):

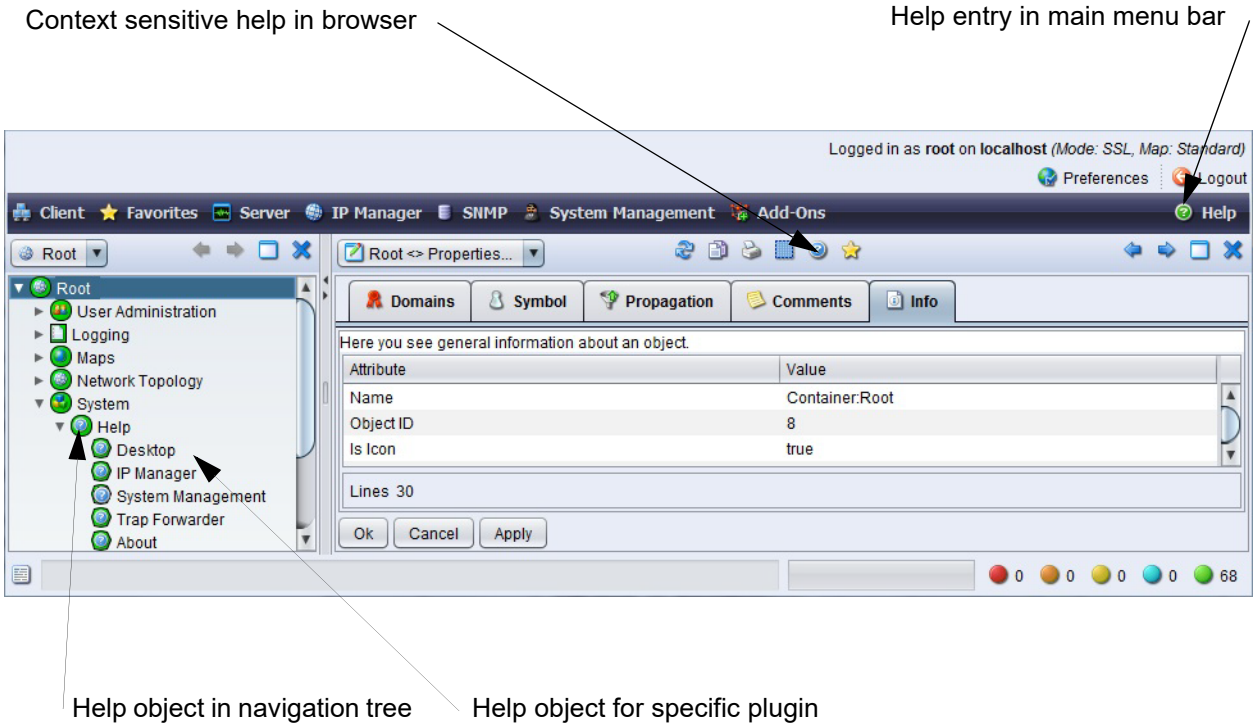


Figure 30 Where to open the Help system

- The main menu **Help** contains entries for all initialized plugins. By using the respective menu entry the help content (TOC) of the respective module will be opened.
  In the navigation tree, the object **System->Help** contains help objects for the plugins. Using the context menu of the respective plugin, the help for the specific plugin can also be opened.
- In addition, the main menu **Help** contains the entry **Overview** which opens an index page that allows access to all english and german user guides (also of non initialized plugins).
- A number of windows provide **window specific help**. If such help is available, the button  (context sensitive help) can be used to open the online help at exactly the point where the specific window is explained. From there the help can be freely navigated.

18 Logging

OpenScape FM Desktop offers a logging facility for external OpenScape FM applications and internal (i.e. OpenScape FM base components) activities. The files are located at

`<OpenScape FM installation directory>/server/logging.`

The following files are created when a OpenScape FM server is installed:

```
activity.log
error.log
trace.log
```

All OpenScape FM specific activities concerning security are recorded in the `activity.log`.

All log files contain several fields (Columns):

- **Time**: time on the OpenScape FM server when the logging entry was written
- **Time Zone**: time zone of the system where the logging application is installed
- **Type**: type of the logging entry (Error, Activity, Trace)
- **Group**: is used to group the different logging entries within a logging type
- **Source IP**: IP address of the system which writes the logging entry
- **Application**: name of the application which causes the logging entry
- **User**: responsible user
- **Mnemonic**: short description of the message

OpenScape FM management applications can write to the three files mentioned above via the RMI interface – that cannot be configured via the OpenScape FM GUI but depends on the implementation of the respective program.

The client will only serve as a tool to manage the log files and to create specific views. A view (like a view in a relational database) is a named defined subset of one of the logfiles which makes it easier to read certain log entries, say from one application only, even though the log file contains a mix of log entries from several applications. Furthermore, the system administrator can manage the access to the views by assigning certain view-specific rights to several users and not assigning them to other users. To learn more about how to assign object rights, please read *Chapter 15, “Access Rights”*.

In order to work with the logging module, open the “Logging” on the root submap (see Logging symbol in *Chapter 13, “Logging Symbols”*). The submap contains three predefined symbols: the `activity.log`, `error.log`, and `trace.log` symbols which refer to the files mentioned above. Each symbol/object offers several operations via its context menu.

- **Show** opens the log file and displays the informations of the fields in columns (*Figure 31*). Like in every OpenScape FM table, you can sort the log file entries with a double click on the corresponding column title. If a line is selected the **Show Description** button will be enabled. Press the **Show Description** button to view the long description for the selected line.

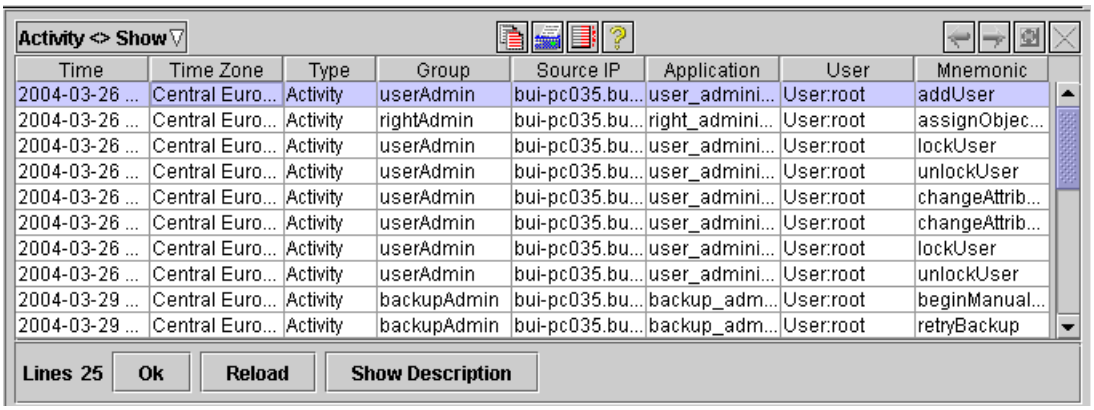


Figure 31 Log File displayed with “Show”

- **Configure** is a menu item which is only available for users of the “Administrator” categories. It lets you set the path to the logfile and define its size parameters. See *Section 18.1, “Log File Configuration”*.
- With **Add View** or **Remove View** you can create/delete a log file view (see View symbol in *Chapter 13, “Symbols and Status Display”*). Read *Section 18.2, “Log File Views”* to learn more about views.

18.1 Log File Configuration

Figure 32 shows the LogFile Configuration GUI which is available for users of the “Administrator” categories. In the **Filename** field, the path to the log file is displayed – the filename cannot be modified. In **Max. File Size**, the maximum file size can be defined, and in **Parts** you can set the number of actual log files which are themselves transparent to the user and form the defined log file.

Figure 32 Configuration of log file

For example, when you define 5 parts for the file activity.log, five files will be created: `activity.log.1`, `activity.log.2`, `activity.log.3`, `activity.log.4`, and `activity.log.5`. For you, the access point in log file management is the file `activity.log` (you do not have to care for the separate parts), and the automatic turn-around will be managed by OpenScape FM. When the maximum file size is exceeded, OpenScape FM will delete the oldest of the log file parts and create a new one. Since the entire file would be removed in that case, it is not possible to set the **Parts** field to “1”!

18.2 Log File Views

When you have created a new view via the context menu of a logfile symbol and **Add View**, you find the symbol of the new view on the log file's submap. Then you can go ahead and configure the view by the help of a filter. As mentioned above, a view is a named subset of information which are contained in a log file. Say, for example, you have a large log file with entries from several users. You could then define a filter to see only the entries from user XY. The filter is defined via the context menu of the view.

- **Show** displays the view, i.e. only the lines of the log file which pass through the filter.
- **Filter** lets a user configure the filter, *Section 18.2.1, "Configuration of Filters"*
- **Configure** is available for users of the type "Administrator" only. Here you can set limitations (*Section 18.2.1, "Configuration of Filters"*) which will be valid for all users. i.e. a user can only restrict but not extend the filter criteria which have been set by root.
- **Remove View** deletes the view.

18.2.1 Configuration of Filters

The menu items **Filter** and **Configure** of the View symbol context menu open the filter configuration GUI (*Figure 33* and *Figure 35*). A filter is defined by the help of Regular Expressions, which can be defined for each field (column) of the log file. Since Regular Expressions are a very complex subject, we cannot provide an exhaustive explanation at that point. However, in the next paragraph, we will try to give you some examples which might occur in every-day administrator life. Please refer to the specific books, e.g. O'Reilly, Jeffrey E.F. Friedl: "Regular Expressions" for detailed information.

Regular Expressions: some examples for search patterns

`.*`	every pattern of characters and/or numbers (default setting for every column)
`^[Ss].*`	every pattern of numbers and/or characters which starts with a small or with a capital s; i.e. you could find the usernames "Smith", "smith", and "salmon" in the username column by the help of a search pattern like User:[Ss]*
`.*User.*`	every pattern which contains the substring "User"; you could for example find all User operations in the Mnemonic field with this search pattern
`.*\.sun\.com$`	every pattern which ends witch the substring ".sun.com"; since the dot has a special meaning in Regular Expressions (one arbitrary character), it has to be masked by a backslash; you could use this pattern for example in the "Source IP" column in order to find all log entries related to machines in the domain sun.com.

There are two filters which are applied one after another: first the Root Filter filters the log file; it can be defined by the root user via **Configure**. Then the User Filter is passed, which can be defined via **Filter**. When the view is displayed, only the lines which have passed through both filters, are to be seen.

18.2.1.1 Filters defined by “Administrator” users

The root user can define filters via the view’s context menu item **Configure**. In the Root Filter Configuration GUI (*Figure 33*) the Regular Expressions for every field in the log file lines can be defined. The Root Filter uses the original entries of the log file. Furthermore, root can decide if the entries are supposed to be filtered in the original version or in a localized version. In order to localize the filter, mark the checkbox **Localized**. Here you can switch on the localization of log file entries, for more details see *Section 18.2.1.2, “Localized Log File View”*

Furthermore, in this GUI you can switch between the view itself and the filter which it used to configure it. That way you can check the results of your filter definition on the fly. Use the buttons **Configure** (in the view) and **View** (in the configuration GUI) for that.

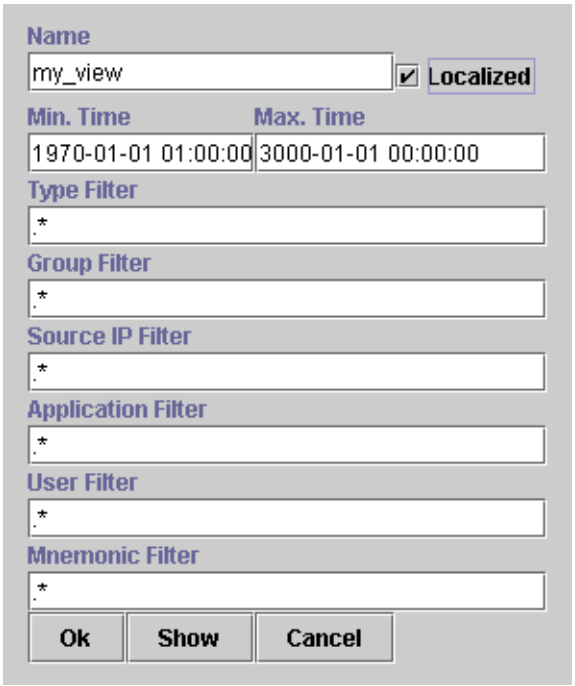


Figure 33 Root Filter Configuration GUI

18.2.1.2 Localized Log File View

The log file entries are not always easy to understand, since they may be written in another language. For that, the logging feature provides localization for log file entries. To enable the log entry localization select the **Localized** checkbox (is selected by default) in the log file view configuration dialog. Press **Show** button to get a localized view of the log file. **Show** is similar to **Show** from **Log File View** menu. For each entry the fields

- Type
- Group
- Mnemonic
- Description

are localized.

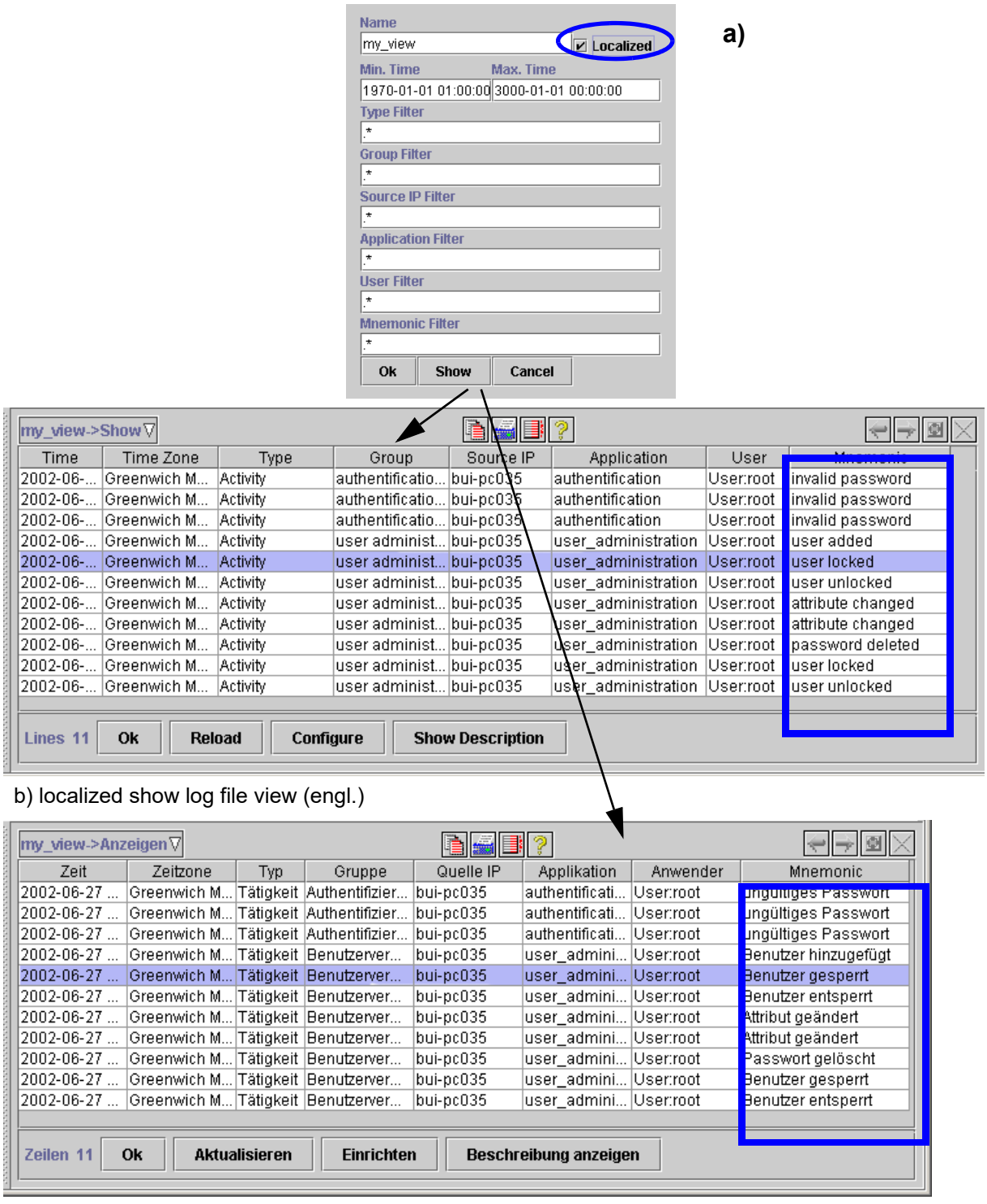


Figure 34 Logfile View browser localized (english and german)

18.2.1.3 Filters defined by regular users

The User Filters always work on localized entries! Say, for example, you only want to read log file entries from the user “Snoopy”, but you are not quite sure if the username starts with a capital letter, you define a filter like the one in *Figure 35*. Of course your user defined filter will only work on the entries that have passed the root defined filter -i.e. when you are allowed to read the respective entries.

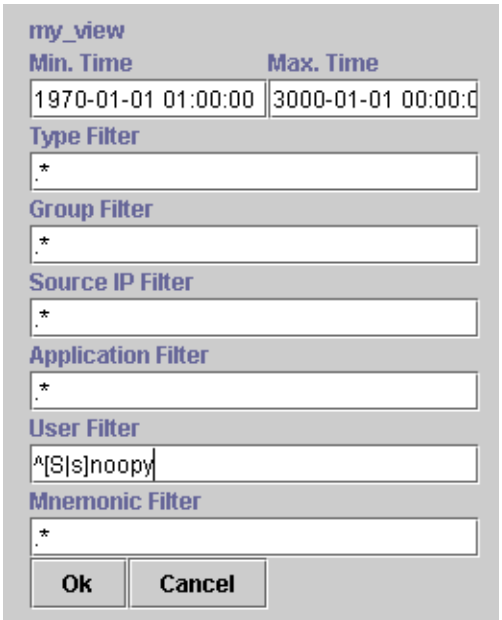


Figure 35 User Filter Configuration GUI

Now the log file will be filtered line by line before it is displayed as view “my_view” , and only the lines which contain the entry for the user “Snoopy” or “snoopy” at the beginning of a line will be displayed.

That way, you can create a view for every selection you need from your log files.

19 Backup and Restore

The OpenScape FM Desktop provides a Backup Management Service which can be used by other OpenScape FM Management applications. The Backup Management Service is provided by the Backup Manager which controls the two basic service operations “backup” and “restore”. The Backup Manager provides an interface allowing OpenScape FM Management applications to register to the Backup Management Service. Hence, only OpenScape FM Management applications which support this interface will be able to participate in the backup service of the OpenScape FM Desktop. A OpenScape FM Management application itself defines which files are object of the backup (for details see documentation of the corresponding OpenScape FM Management application). The Backup Management Service periodically asks registered OpenScape FM Management applications to perform a backup. A OpenScape FM Management application is responsible for the creation of a consistent backup of its data.

Important Note:
If the System Management Plugin is used (see separate User Guide) the configuration of the System Management Agents will not be stored by the general backup mechanism.
In this case a Backup Monitor has to be configured for each agent if desired.

The Backup Manager controls the services “backup” and “restore”.

Backup Operation: There are manual and automatic backups supported. Manual backups are initiated by a user with the right Backup Administrator (in the following such a user will be called Backup Administrator). Automatic backups are performed periodically by the Backup Manager. A registered application will be asked by the Backup Manager to backup its data. The application uses the backup interface to perform the requested backup. The Backup Service will save the data to a directory which can be configured in the OpenScape FM GUI by a Backup Administrator. When an application could finish a backup operation successfully, it signals this to the Backup Manager and logs this in the Activity Logging, see *Chapter 18, “Logging”*.

Restore Operation: A restore operation has to be initiated by a Backup Administrator. The concerned OpenScape FM Management application will be notified by the Backup Manager to perform a restore data operation. When an application could finish a restore operation successfully, it signals this to the Backup Manager and logs this in the Activity Logging, see *Chapter 18, “Logging”*.

Two scopes for backup and restore operations are available which have to be distinguished:

1. operations and configuration tasks done for the Backup Manager and

2. operations and configuration tasks done for an individual Backup Application.

The Backup Manager level can be used for the global administration. But it is also possible to configure only the default parameters for new Backup Applications.
A manual backup operation on the level of the Backup Manager will always be performed for all registered and managed Backup Applications.

All settings on the level of an individual *Backup Application* will overwrite the default parameters for this Application. That way an individual configuration for each existing Backup Application will be possible. All operations on the level of an individual Backup Applications will only effect this application.

Initially only the OpenScape FM Desktop is registered for the Backup Service (OpenScape FM Database). The OpenScape FM Desktop application is per default registered and always available (except the application is in the state unmanage, see *Section 19.5, “Unmanage/Manage”*) for backup and restore operations.

Hint:
A backup for the OpenScape FM Desktop application covers the OpenScape FM Desktop database, which includes all network management-relevant information and Enterprise MIB definitions; it does not include any images (e.g. background images) and licenses.

19.1 Backup Manager

The Backup Manager icon appears under the **Root->System->Server->Administration** icon. It is a container object for all registered Backup Applications.

The Backup Manager icon offers a context menu which is divided in two parts; the upper part contains the general OpenScape FM menu items, whereas the lower part provides several object specific menu items. In addition identical menu entries with identical functionality can be found in the main menu **Server->Administration->Backup Manager**:

- **Edit Backup Parameters...** here the default backup parameters for all **new** registered Backup Applications can be set, see Section 19.2, “Edit Backup Parameters”.
- **List Backups...** here a browser opens where all backups of **all registered and managed** Backup Applications are listed. Over this browser also a backup for Backup Applications can be restored. (*Section 19.4, “List Backups and Restore Browser”*)
- **Manual Backup** here a backup for **all registered and managed** Backup Applications can be started, see *Section 19.3, “Manual Backup”*.

With a double click on the Backup Managers icon a submap opens which contains all registered Backup Applications. A Backup Application which is registered and ready for backup and restore operations is represented by a green colored icon, other states will be described in *Section 19.7*. The icon provides the same menu items as the Backup Manager and some additionally:

- **Edit Backup Parameters...**, here the backup Parameters for **this Backup Application** can be configured, see Section 19.2, “Edit Backup Parameters”.
- **List Backups...**, here a browser opens where all backups of **this Backup Application** are listed. Over this browser a restore operation can also be started, see *Section 19.4, “List Backups and Restore Browser”*. Individual backups can also be deleted from here.
- **Manual Backup**, here a Manual Backup can be started for **this Backup Application**, see *Section 19.3, “Manual Backup”*.
- **Cancel,** [this menu item is only available when a restore or a backup is active.] An active process can canceled over this menu item, see *Section 19.6, “Cancel”*.

19.2 Edit Backup Parameters

To configure the Backup parameters for backups it is necessary to open the **Edit Backup Parameters...** dialogue.

Note:
The dialogue can be opened in the Backup Manager context and in the Backup Application context. When the dialogue is opened in Backup Application context the parameters **only apply to this application**.

Backup Manager context:

- Main Menu **Server->Administration->Backup Manager->Edit Backup Parameters...**
- Backup Manager symbol->**Edit Backup Parameters...**

Backup Application context:

- Backup Application symbol->**Edit Backup Parameters...**

The structure and functionality of the dialogues of the Backup Manager and the Backup Application are nearly identical. But the scope of the configured parameters is different.

- In the Edit Backup Parameters dialogue the time interval for the execution of automatic backups can be selected. For example, it can be specified that a backup shall be performed every day at a certain time, every n-th day at a certain time, or every n hours (n minutes). To set up a specific backup configuration, the time interval, which is supposed to elapse between backups, has to be defined in the field **Backup every** . Subsequently the time unit (minutes, hours or days) for the time interval of the automatic backups has to be chosen. The **Next start at** label indicates the date and time of the first backup and the start time of the interval timer. By double-clicking the date- and time field the backup time can be set by using a special user interface.
- **Number of Backups kept**, specifies the number of backups to be retained. If a new backup is created and the “number of kept Backups” is reached, the oldest backup will be deleted. For example, a backup is performed every day and the number of kept backups is “5”, five backups will be kept; when the sixth backup is created the oldest one will be deleted. When the number of kept Backups will be changed to a smaller value, only the oldest backup will be deleted after a backup. For example, if five backups have already been created and the number of kept Backups is changed to “2“. After a backup only the oldest one will be deleted and still five backups will be found in the backup directory.
- **Number of max. Retries**, specifies the maximum number of retries a backup or restore operation has to be performed till the operation will be stopped. For example if the Backup Application temporarily is not available during a backup operation and several files are already transmitted, the Backup Manager initiates the transfer of the last backup file until the backup or restore succeeded or until the number of retries is reached.
- In **Backup to** the directory can be specified where the backups shall be stored. The path can be entered manually or the Remote File Chooser (via **Browse...**) can be used to select a directory.
  Two cases have to be distinguished:
  - **Case 1:** If the Backup to directory is specified via the Backup Manager it points to a directory which will contain subdirectories for each registered Backup Application. These subdirectories will contain the backup files of the applications. Per default the “Backup to“ directory is set to `/<OpenScape FM Installation directory/server/backup/.`
  - **Case 2:** If the Backup to directory is specified in the context of a backup application it points to a directory keeping all backup files of this application. Default is the “Backup to“ set in the Backup Manager context incremented by the subdirectory for the corresponding Backup Application.

- The field **Free Disc Space Requirement for Backup** specifies the minimum amount of space that must be available on the selected file system before a backup can be started.
  If there is not enough space available when a backup is initiated, an event with the status *Major* is generated for each component to be saved and a corresponding message is displayed. Regardless whether it is an automatic or manual backup.
- With the checkbox **Automatic Backups Enabled**, the automatic backup can be switched on or off. This checkbox is per default deactivated.
- The checkbox **Change configurations for all registered clients** is only available in Backup Manager context. It indicates, if the configured parameters will be set for **all** Backup Applications (already registered and new) or only for **new** registered Backup Applications. If the checkbox is checked the configured parameters will overwrite the existing configuration of **all** registered Backup Application after pressing the OK button. If the checkbox is not checked the configured parameters will only be used for **new** registered Backup Applications.
  The checkbox is checked by default.

By pressing the **Ok** button the read and write permissions of the **Backup to** directory will be checked and the path modification and the other values will become valid.
In Backup Application context the new directory will be scanned for backup entries belonging to this application.

Important Note: [Backup Manager]
If the dialogue is opened in Backup Manager context and the checkbox **Change configurations for all registered clients** is checked, pressing the **Ok** button will (by right configured values) always overwrite the values of all Backup Applications. This applies also if no changes have been made in this dialogue.

Note: [Backup Application]
If a backup or restore operation is active while changing some parameters in the GUI, all changes will become valid after the current operation has been finished!

If the directory restricts access to read-only:

- The appropriate application symbol label is surrounded by stars (Figure 36).
- The **Manual Backup** action is not available via the applications context menu.
- An automatic backup will not be performed.
- But it is still possible to start an restore.

When the directory gets write-rights later on, the Backup Application has first to be reinitialized by setting it to Unmanage and then back to Manage, see *Section 19.5, “Unmanage/Manage”* or a manual backup has to be done for this application, see Section 19.3, “Manual Backup”. This will also remove the stars from the application label.

Figure 36 Backup Application: read-only directory

19.3 Manual Backup

Independent of the automatic backup mechanism a manual backup can be performed. As same as the Edit Backup Parameters dialogue the manual backup can be done on the Backup Manager and on several Backup Applications.

To start an instant backup manually for a **application**, the menu item **Manual Backup** in the applications context-menu has to be used. A backup for **this Backup Application** will be performed immediately and will be saved in the selected directory (see Section 19.2, “Edit Backup Parameters”).

With the menu item **Manual Backup** in the **Backup Managers** context-menu or be using the same item from the main menu **Server->Administration->Backup Manager** a user can start an instant backup for **all registered and managed** applications.

The page that initially opens shows a list of all components known to the backup. There is a leading green check mark for components that are registered as active and that have a connection to the OpenScape FM Server.

Clicking the **Manual Backup** button starts the actual backup processes.

The backups will be performed immediately and will be saved in the corresponding directory (see Section 19.2, “Edit Backup Parameters”).

A displayed hourglass icon stands for a currently running backup process, a double green check mark for a successfully completed process, and a red cross for a process aborted with an error.

An error may occur, for example, if the connection to an agent to be stored is lost, or if the storage space for the backup file is insufficient.

When a backup is started, the state of the symbol changes to testing (indicating the user that a backup is active, see *Section 19.7, “Application Status”*). Additionally the icon label of the Backup Application displays some useful transfer statistics information, see *Figure 37*.

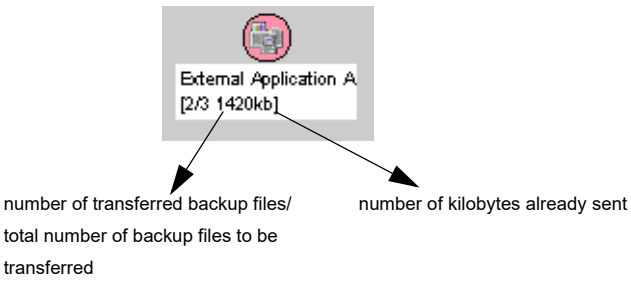


Figure 37 Backup Application: Active Backup

Note:
Only when the application is in status Normal (Section 19.7, “Application Status”), the menu-item Manual Backup will be accessible, i.e. no backup or restore operation is active

19.4 List Backups and Restore Browser

This chapter describes how to list backups (*Section 19.4.1, "List Backups"*) and to restore backups (*Section 19.4.2, "Restore"*). Like the Edit Backup Parameter Dialogue these actions can be performed in Backup Application context and in Backup Manager context: For example the List Backups action will in Backup Application context only list the backups of the Application, in Backup Manager context it will list all backups of **all** Backup Applications.

19.4.1 List Backups

There are two possibilities to list Backups: for all managed and registered applications and for one application only.

- To list the backups of **all** registered and managed applications choose either the **List Backups...** menu item of the **Backup Managers** context-menu or in the main menu **Server->Administration->Backup Manager**.
- To list the backups of **one application** choose the menu item **List Backups...** of the **corresponding application**.

In both cases an Info Browser (*Figure 38*) will be opened, listing all backup entries. In **Backup Application** context additionally in the top area the name of the application and the path of the corresponding backup directory are displayed.

The Info Browser shows the following information:

- **Application:** The name for the registered Backup Application. [This column is only in Backup Manager context available.]
- **Status:** The status of a backup or restore operation. If a restore operation is busy, "restore active" is displayed. If a restore or a backup operation has been accomplished, "finished" is displayed.The state "corrupt" indicates that the corresponding backup is not complete or readable.
- **Time (generated):** A time stamp documenting the time of the generation of this backup entry and the information about the backup operation. A backup (entry) is generated by a manual or automatic backup operation.
- **Count Files:** The number of data files backed up for this application.
- **Total Size (Kb):** The total storage capacity (number of kilobytes), all backup data files claim.
- **Transmission Time:** Indicates the execution time of the related backup operation.
- **Last Restored Date (generated):** The latest date when the backup entry was object of a restore operation, see *Section 19.4.2, "Restore"*.

With the **Reload...** button updates the browser content and state of the backup/restore activities.

The **Restore** button is used to start restore operation(s). This will be described in detail in *Section 19.4.2, "Restore"*.

When the List Backups Information browser is opened via a **Backup Application**, the browser offers additionally the buttons **Choose Directory...** and **Set Default Directory**. The button **Choose Directory...** allows a user to switch to another directory in order to list the backup entries of that directory. Pressing the **Choose Directory...**

button a Remote File Chooser opens for selecting another directory from the file system of the OpenScape FM server. Having selected a directory via the Remote File Chooser all backup entries of this application stored in the new directory are listed. It is also possible to restore a backup from this directory. This directory will be shown in the List Backups Info Browser as long as it will be switched back to the default directory.To switch back to the default directory, the one which was set before via the Edit Backup Parameters GUI (*Section 19.2, "Edit Backup Parameters"*), press the **Set Default Directory** button. This button is only available, if another directory over the **Choose Directory...** button is chosen.

Note:
The actions of the buttons **Choose Directory...** and **Set Default Directory** will not change the backup parameters of the corresponding application.

19.4.2 Restore

The List Backups Info Browser offers the button **Restore...**. This button is to be used for the restore operations.

When the List Backups Info Browser is opened in **Backup Manager** context it is possible to perform a restore operation for multiple applications at the same time. In the **Backup Applications** context only one backup can be restored.

In order to perform a restore for one application a backup entry of this application has to be selected and the **Restore...** button pressed. Then the "Restore Information" window appears warning the user about a possible restart of the application after the finishing the restore operation. In case the restore shall be performed the **Ok** button shall be pressed, otherwise the button **Cancel.**

Note:
The successful loading of backups is only ensured for backup files that have been created with the current or the previous OpenScape FM release (e.g. in the case of OpenScape FM release 4.3 for backup files created with release 4.3 or 4.2). If older backup files cannot be imported successfully, it may be necessary to use older OpenScape FM Server releases as intermediate steps.

If in the Backup Manager context more than one backup entry for the same application has been selected, a warning message will appear in the Message Log Window. To resolve this error the invalid backup entries have to be deselected, first.

After acknowledging the "Restore Information" window, the corresponding application symbol label changes as shown in *Figure 38*. The label of the Backup Application icon displays helpful information (transfer statistics) like the number of files already been restored and the number of bytes already been sent.

Note:
The button **Restore...** will only be provided when for the application no backup or restore operation is active!

When an application refuses a restore request the Backup Manager remembers this fact and resumes this request later on after a reregistration of the application.

Warning:
After a successful restore of the OpenScape FM Database, the OpenScape FM server automatically restarts via the Startup Manager (see *Section 4.1, "Starting the Server"*) with the restored OpenScape FM Desktop database files.

When the List Backups Info Browser is opened in **Application** context the browser offers additionally the buttons **Choose Directory...** and **Set Default Directory**. Over the button **Choose Directory...** another directory for the subsequent restore operations can be chosen. To reset the directory to the value configured in the Edit Backup Parameters dialogue select the button **Set Default Directory.**

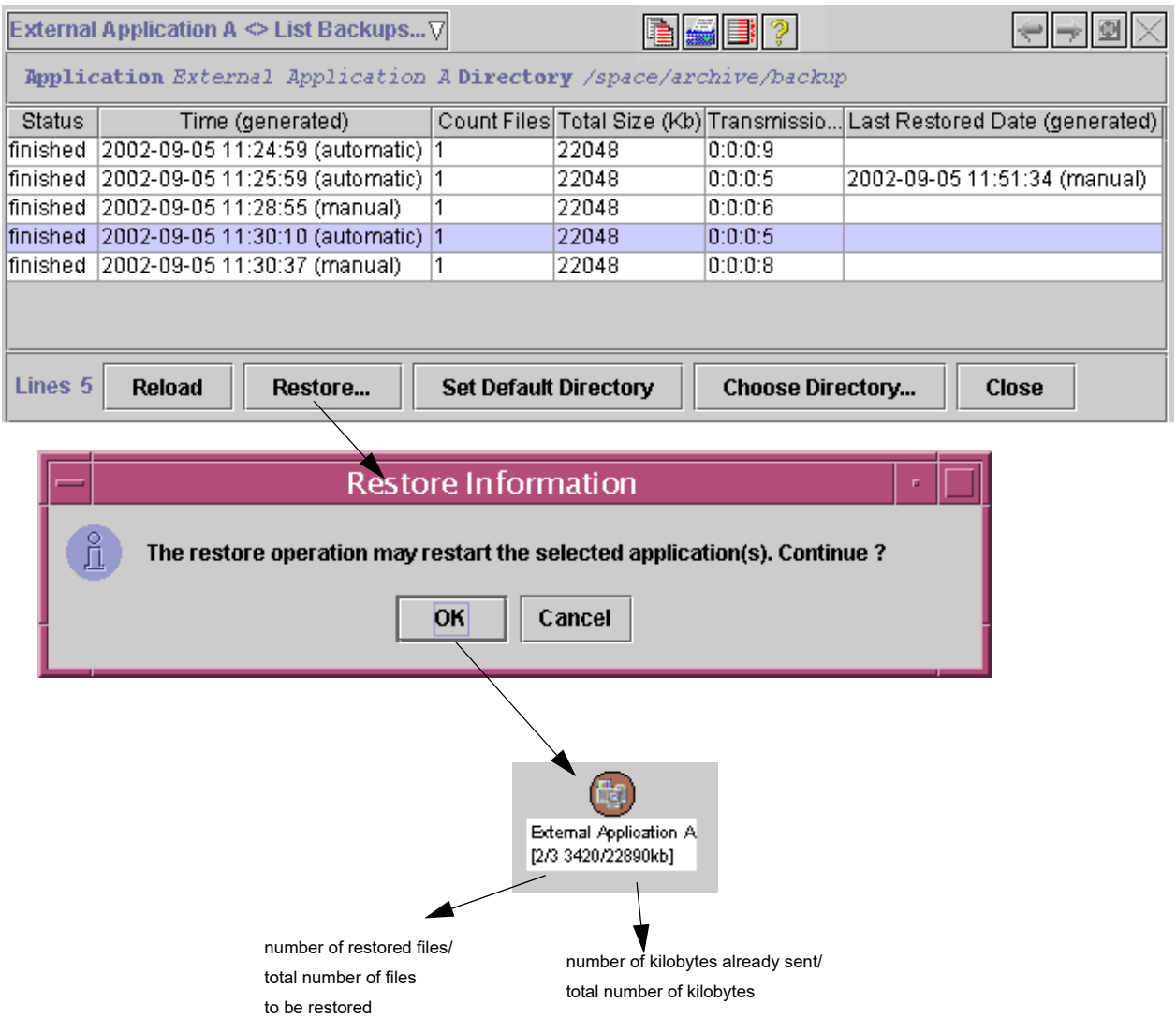


Figure 38 Backup Application: List Backups and Restore

In order to restore old data which were generated without the Backup Manager, i.e. were generated with older OpenScape FM versions, you have to stop the Server and execute the file `startRestore.exe` (Windows) / `startRestore` (Unix) which is to be found in the installation directory. Then select the "base" file you would like to recover the data from. It is not possible to select any file types other than "base" files! The restored database will then be activated and the Server will be started. The old database will be saved as "database_old".
Since the server is already running, you can start the client afterwards and start work with the restored data.

19.5 Unmanage/Manage

Using the menu item **Edit->Unmanage** a Backup Application can be suspended in a handy way from the backup process. An unmanaged Backup Application will be displayed (see *Section 19.7, "Application Status"*) all specific menu items will no longer be available.

An unmanaged Backup Application can be set back to a managed application by using the menu item **Edit->Manage**. The destination "Backup to" backup directory from *Section 19.2, "Edit Backup Parameters"* will then be scanned for existing backup entries and the Backup Applications status will be recalculated immediately.

Important Note:
In case of a busy application will be set to unmanaged, the active backup operation will be canceled.

19.6 Cancel

If a backup or restore operation is active, it can be canceled by selecting the menu item **Cancel**. The symbol status for the application changes to normal and the symbol label transfer statistics will disappear.

19.7 Application Status

The appearance of an Backup Application symbol depends on the type and the state of the application it represents. States are represented by colors. When an error occurs the symbol of the corresponding Backup Application starts to blink in order to notify the administrator that a new event has arrived.

There are seven different states for Backup Applications:

Unmanaged → light brown
This application is still registered in the database, but there are neither backup nor restore operations available. If the Backup Application was manually unmanaged by a user, it will get this status.

Testing → salmon
The status „testing" indicates a busy backup operation for the application.
Start of blinking in this state means that a backup operation was refused by the Backup Application. The corresponding event in the Event Browser shows the number of refusals for this application.

Disabled → dark brown
The status „disabled" means that a Backup Application was unregistered manually.
Neither backup nor restore operations are available.

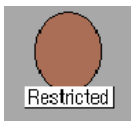
	Restricted → brown The status „restricted“ indicates a busy restore operation for the application. Start of blinking in this state means that a restore operation was refused by the Backup Application. If the application re-registers the next time, the backup manager will remember the refused backup entry and automatically starts restoring the data.
	Normal → green The Backup Application is in a normal operational state. It is ready for backup and restore operations.
	Major → orange The last backup or restore operation failed and was interrupted due to an I/O error or network problems. Start of blinking in this state means that there are no permissions to the destination backup directory (see *Section 19.2, “Edit Backup Parameters”*) for this application.
	Critical → red The Backup Application is no more responding. Start of blinking in this state informs the user that the Backup Application is not responding and no more available. This status is only set, if the number of retries (see *Section 19.2, “Edit Backup Parameters”*) is exceeded.

Table 4 Status of Backup Applications

19.8 Deleting a Backup Application

When a Backup Application is no longer in use, it can be deleted from the Backup Managers submap by using the menu item **Edit->Delete Object**.

Warning:
When a Backup Application has been deleted from the Backup Managers submap the created backups in the configured directory will be deleted, too.

To use the Backup Manager service again the Application has to reconnect to the Backup Manager.

Note:
When the OpenScape FM Database object will be deleted from the Backup Managers submap all backups will be deleted, but after a restart of the OpenScape FM Server it will register itself again.

19.9 Logging of Backup operations

All Backup operations will be logged in the OpenScape FM Activity Log. Errors will be reported in the Error Log. The Logging component is explained in detail in *Chapter 18, “Logging”*. The operations are performed under the “backupAdmin” account - thus the corresponding log file entries can be retrieved by looking the “backupAdmin” in corresponding “Group” of the log files.

20 Time Schedule

Time Schedules also named Time Filters are calendar-like objects which are used to define appointments and time intervals. They are used for various purposes in OpenScape FM, for example:

- **Mobile Alarm Reaction (MAR):** Definition of time intervals within which reactions should be triggered, e.g. sending Email during working time and sending SMS outside working time
- **Event Browser:** Show only events which were received within a particular time period
- **Report Manager:** Schedule automatic report generation
- **Event Correlation Engine (ECE):** Time-dependent filtering of events
- **Event Correlation Engine (ECE):** Scheduling of planned maintenance intervals

20.1 Configuration

The configuration dialogue for a time schedule can be invoked from every place where time schedules are used, e.g. in the MAR or the Report Center. A central overview about all configured time schedules can be opened by selecting **Server->Time Scheduler...** from the main menu bar.

The Interface for time filer configuration is calendar-based and offers five different views: Year, Month, Week, Day and List. Each view shows the existing appointments for the displayed time range (e.g. the current month). An exception is the list view: It provides an overview about all appointments of the current time schedule, independent from any time range.

20.1.1 Single Appointment or Time Interval

In the year, month, week and day view, new appointments can be added or edited by a right mouse click and selecting **Add Appointment** or **Edit** from the context menu. This opens a dialog where the appointment can be configured by specifying the following attributes:

- **Description**
  An arbitrary description for the appointment or time interval
- **From**
  The start time for the appointment or time interval in the format "`dd.mm.yyyy - hh:mm`" in 24h format
- **To**
  The end time for the appointment or time interval in the format "`dd.mm.yyyy - hh:mm`" in 24h format

This is the most simple way to define a new appointment. The other attributes can be left as they are if just a single appointment shall be defined. Further configuration options are described in the following sections.

20.1.2 Series of Appointments or Time Intervals

To define a series of appointments, i.e. appointments which are repeated periodically, the following additional attributes have to be configured:

- **Period**
  The repetition interval for the appointment: Day, Week, Month, Year or Once (default for single appointments)
- **Repetition end**
  The date when the repeated appointment series will end
- **Multiplicator**
  The multiplicator defines the gap between two subsequent repetitions. Example: The period "Day" and a multiplicator of "2" will repeat the appointment every two days.

To provide a better orientation within the overviews, the appointments belonging to the same series are displayed with the same color coding (background and border color).

20.1.3 Exclusion of Particular Days from a Series of Appointments

Exclusion is used when an appointment shall be repeated periodically but there are some exceptions when the appointment shall be skipped. Such exceptions are defined as follows:

- **Exclude**
  An exclusion date can be entered in the format "`dd.mm.yyyy`". By adding the date to the exclude list, the series of appointments will be skipped for this date. Multiple exclude dates can be added.
- **Interval Type**
  The interval type is used to tag an appointment or time interval as "In Service" or "Out of Service". This tagging is used by some components like the report manager to distinguish between different intervals. For example, you can create reports based only on events which were received during service times.

If multiple appointments with different interval types are overlapping, the "Out of Service" appointment has priority. Example: If there is a "Service" interval defined from 12:00 to 14:00 and an "Out of Service" interval from 13:00 to 15:00, a report based on this time schedule for events received "in service time" will only include events received between 12:00 and 13:00.

Hint:
If an "Out of Service" interval does not overlap with at least one "In Service" interval, it has no effect at all.

20.2 Deletion

Appointments can be deleted at the year, month, week or day view by a right mouse click on the appointment and selecting the appropriate delete function. If the appointment belongs to a series, either the whole series can be deleted or just the single appointment.

21 Startup Manager

The OpenScape FM Startup Manager is included in the OpenScape FM installation.

On Windows systems it is available as a service named "`OpenScape FM Startup Service`" and can be controlled via the Windows Service Manager.

On Unix systems it is a daemon and can be started or stopped via the executables `<OpenScape FM installation directory>/startStartupService` (start) and `<OpenScape FM installation directory>/stopStartupService` (stop). After a reboot the rc-script named `S99OpenScape` will be invoked to start the system automatically.

The Startup Manager is used to start, stop and restart the locally installed OpenScape FM server. If an unexpected termination is detected, the Startup Manager will automatically try to restart the OpenScape FM server for a limited number of times, per default twice. If the number of automatic restarts is exceeded, the Startup Manager does not restart it until the whole Startup Manager will be restarted, or the administrator restarts the OpenScape FM server manually.

By default, the Startup Manager starts the OpenScape FM server and other processes needed by the OpenScape FM (e.g. Tomcat, Java DB). However, it can be configured to start other services, too.

Important Note:
Just like for the OpenScape FM server, the number of restarts after an unexpected termination is individually counted per service for all other automatically started services. If the maximum number of allowed restarts is reached for *one of the services*, the Startup Manager and thus also the OpenScape FM is automatically shut down to avoid an inconsistent overall system state.

Important Note:
The counters of the individual monitored services are also not reset after a successful restart of the service. The reset is only performed when the Startup Manager is restarted.

Note:
The monitored services are started with the Java option `ExitOnOutOfMemoryError`. This option should usually terminate a service in an orderly manner the first time an OutOfMemory error occurs, in order to prevent uncontrolled behavior due to lack of memory.

The configuration of the OpenScape FM Startup Manager will be discussed in the *Technician Guide*.

The services controlled by the Startup Manager can be categorized in three types which differ in the way they are handled: conform, non conform and daemons.

- **conform**: These services support the OpenScape FM Startup Manager. The service will be notified about stop or shutdown by the OpenScape FM Startup Manager. Conform services can inform the Startup Manager about their state in the process life-cycle. Depending services will be started after the required services have reached the state "running".
- **non conform**: These services do not support the OpenScape FM Startup Manager. The services will be killed on stop or shutdown. The state of the started service is considered "started" immediately or after a defined time-out.

- **daemon**: These services will just be started and not be taken care about. The state of the started service is considered “started“ immediately or after a defined time-out. Stop and restart action will not work for these services.

The Startup Manager allows to define dependencies between services. When a dependency is defined for a particular service all services on which this service depends on are started first. When a service with dependent services is stopped, the dependent services will be stopped before. Exceptions are daemon services and services that are terminated unexpectedly. In this cases depending services will not be stopped.

The OpenScape FM Startup Manager can be monitored using the OpenScape FM user interface, see *Section 21.1, “User Interface”*.

21.1 User Interface

After the start of OpenScape FM there will be a Startup Manager icon on the Server submap (**Root->System->Server->Administration->Startup Manager**). Below the Startup Manager icon an icon symbolizing the OpenScape FM service will be located. When the Startup Manager has been configured to start other services, there will be icons for those services, too. The context-menu of the Startup Manager icon offers the option **List Services...** which opens a browser showing the properties of all services. This menu item will also be found at the main menu **Server->Administration->Startup Manager->List Services** in the main menu bar.

In the List Services browser the following properties for services will be displayed:

- **Service Name:**
  The name of the service (its ID).
- **Status:**
  The current status of the service see *Section 21.2, “Status of a Service”*.
- **Status Message:**
  An optional status-message which can be provided by the service.
  Only the last message is displayed.
- **Uptime:**
  The Uptime of the service.
- **Startup Type:**
  The startup-type of the service.
- **Dependencies:**
  Dependencies to other registered services which have to be considered for start/stop of the service.

The List Services browser shows one registered process by row. With the Buttons **Start Service**, **Stop Service** and **Restart Service** you can start, stop or restart the selected process. The Button **Reload** refreshes the displayed data.

Warning:
When the OpenScape FM process is marked and the button **Stop Service** is clicked, the OpenScape FM server the user is logged in will be stopped and all clients will be closed. We would therefore recommend that the system should be configured in such a way that not every user could use this dialogue, see *Section 21.3, "Rights Configuration"*. Make sure that you really want to stop the server you are currently logged in.

The icons of the services offer a context menu which contains the following menu items:

- **Properties...**
  Displays a browser with all properties related to the selected service.
- **Stop Service**
  [Only available when the service is in "*running*" state]
- **Restart Service.**
  [Only available when the service is in "*running*" state]
- **Start Service**
  [Only available when the service is in "*terminated*" state]

21.2 Status of a Service

A service can have different states which are explained below:

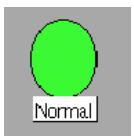
Severity	Status	Description
	starting	The Startup Manager is trying to start the service.
	no start message	The service has not reported running state within the startup timeout [conform services only]. For more information about timeouts please refer to the *Technician Guide*
	running	The service was be started successfully and is now running [conform processes only]
	started	The service is started. [non_conform services and daemons]

Table 5 Status of a service

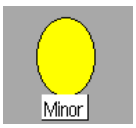
Severity	Status	Description
	shutting down	The service is shutting down.
	stopped	The service has reported that it is stopped. [conform services only]
	terminated	The service is no longer running.
	never run	The service could not be started, because of problems with the command execution or the configuration file.
	unresolved dependencies	Configured dependencies could not be resolved

Table 5 Status of a service

21.3 Rights Configuration

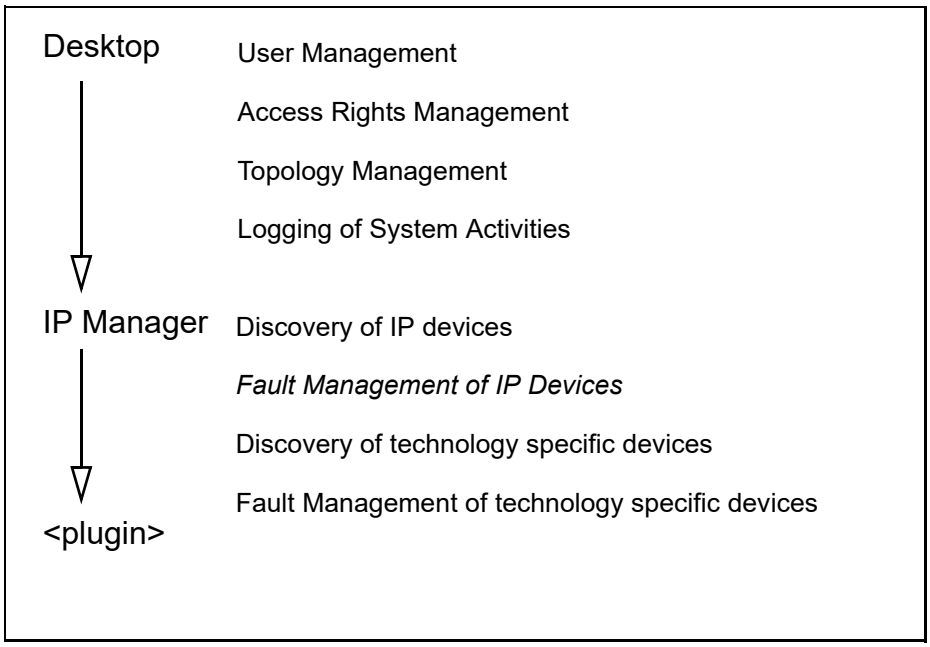
The Startup Manager is a powerful tool. Therefore, it should be secured that not every user has full access to the user interface of the Startup Manager. To restrict the capabilities of the user interface, the access rights of the Startup Manager are used, See *Chapter 15, "Access Rights"* to learn more about assigning rights. The Startup Manager offers two predefined compound rights:

- **Base->Startup Administrator** allows the user, to stop, start and restart the service for which this right has been granted.
- **Base->Startup Operator** allows to view the properties of the service, only.

22 Plugin Modules

22.1 Plugin Modules and Their Interaction

So far you have licensed the OpenScape FM Desktop and possibly initialized plugin modules in the desktop mode. That won't take you far in managing your network – therefore, before we go on explaining the Desktop functionalities, we will give you a short overview over the different OpenScape FM modules and their interaction.



First of all, there is the Desktop, which provides the user interface for your network management. It also provides the basis for several plugin modules which need the Desktop to manage users, to control access rights and to display the network topology.

The IP Manager Plugin is initialized automatically during the initialization process. The IP Manager serves several purposes: it performs the IP discovery, which is the basis for the entire network management, and it can accomplish fault monitoring for IP devices.

Every **plugin** also serves several purposes:

- It provides the functionality to access the technology specific information (i.e. to read the corresponding MIB or to work with the specific HTTP/XML requests/answers).
- It adds the functionality to discover devices which are specific for the <plugin type> environment.
- It processes fault information from the respective devices.

23 Troubleshooting

When problems arise during the work with the Server and/or the Client there are several possibilities to see what kind of error has occurred.

23.1 Log Files

Startup Manager:

The directory `<Installation directory>/startup/log` contains a log file for each monitored service. The OpenScape FM logfile is named `OpenScapeDesktop.log`.

Server:
The files server.log and java.log are protocols of the Server. You can analyze these files or show them to a OpenScape FM specialist to understand the reasons for errors.

The maximum size of the logfiles can be configured using the main menu entry **Server->Administration->Server Properties** and opening the page **Server Process Parameters** (see *Section 6.2*).

The file `trap.log` reports all incoming SNMP traps on the default port 162.

The log files can be viewed from within the client by using the main menu entry **Server->Administration->Debug Options**. For this purpose, a log file must be selected on the debug page using the menu below and the button Display Logfile has to be clicked.

Client:
The first line of the Message Log (see *Section 5.10, "Message Log"*) shows you where the client log file is located.

23.2 Log and Debug Configuration

To provide a simple way to configure debugging and logging options the OpenScape FM Client offers a log and debug configuration utility. This utility can be used to switch server-side debug and log messages on and off separately for each initialized plugin. The files required by the product support to diagnose problems can here also be automatically combined into an archive file.

The log and debug configuration utility window can be opened via the menu item **Server->Administration->Debug Options...** to be found in the main menu bar of the OpenScape FM Client.

The column **Category** displays a list of all initialized plugins. The columns next to it show the current log and debug status. If a checkbox is checked the corresponding option is active for this plugin.

In order to set the option for debugging select one or more plugin(s) in the list, click on the checkbox for debugging to the right of the list and press the  button. The change is shown immediately after the  button is pressed:

- in the column **Debugging** the corresponding checkbox get checked and
- from now on extended debug information will be written to the server logfile.

To get extended logging information in your server logfile, you have to do the same steps as for debugging but you have to select the logging checkbox instead.

The dialogue supports multiple selection, therefore an option (even both) can be set for a selected set of plugins in one operation. If you want to set an option for all plugins, select all shown plugins and click on the wanted checkbox.

The logging and debugging options are independent from each other, therefore logging **and** debugging can be active for a plugin at the same time.

The **Show Log** button will open an info browser displaying the server logfile. Here, you can reload the info browser via Reload and close the info browser via **Close**.

The button **Download SNMP Trap Tool** opens the Test Trap Tool (see *Section 23.6*).

The **Close** button closes the log and debug configuration utility window.

The button **Collect Diagnostic Information** can be used to automatically create a logging archive file that summarizes the collected data required for a diagnosis.

In the window opened by the button, a description of the error can be entered. In addition, it can be selected whether the current thread status, a current memory dump and/or all database files of the server process should be added to the archive. It can also be specified whether the archive should be encrypted and protected by a password. If the **OK** button is pressed, the archive file is created and the user is prompted by a file browser to select a storage location and name.

By using the script file `diagnose.bat` (for Windows systems) or `diagnose.sh` (for Linux systems) the creation of a logging archive file can alternatively be started. These can be found below the OpenScape FM installation directory within the container `diagnose` and have to be started with Administrator rights.

Depending on the selection of parameters, the logging archive files can also be used to create and archive database, heap and thread dumps. The content of a file located within the same directory and with the name `description.txt` can be added to the archive to provide a problem description file.

The script files, in contrast to the method described above, can also be used, if the OpenScape FM process has been stopped.

The debug/log configuration is stored persistently and will survive server-restarts. Therefore it is possible to get debug information during the server startup phase as well as at runtime.

Enabling debug and log options can result in performance losses. Enabled debug and log options must be switched off manually.

It has to be kept in mind that the debug info that gets written is not intended to be analyzed by the user. The output contains technical information about server-internal functions and should be send, if requested, to the third level support or the development team for debugging purposes.

23.3 Server Information

In cases where problems with the OpenScape FM server occurs and the third-level support has to be consulted additional server information has to be known. For that the OpenScape FM Desktop offers for users with Administrator rights an info browser showing all relevant information. This info browser is opened via the main menu item **Server->Administration->Server Properties**. It is located on the page **Info**.

23.4 Browser Configuration

When your web browser cannot load the Client, take a closer look at your browser configuration. The proxy settings might be wrong. When the client and the server are located in the same network you usually do not need a proxy at all. So you can set “no proxy for... “ <Server IP>.

23.5 Address and Name Resolution

When a client cannot find the Server, there might be a general problem with the name resolution in your network. Has the DNS been set up correctly? If you do not use DNS the hosts file on each machine has to be kept up-to-date.

Another problem might arise when you work with the Microsoft Internet Explorer: when you have entered the address of the OpenScape FM server with the correct IP port but without a slash, sometimes the applet cannot be loaded. Thus use the following URL:
http://<name or IP of Server>:3043/

23.6 Test Trap Tool

To be able to report current problems promptly, the OpenScape FM evaluates incoming SNMP traps. If these traps are lost, problems are often only detected at a later time by active polling of the OpenScape FM.

If it is suspected that the evaluation of traps by the OpenScape FM might be disturbed, this can be checked with the aid of the **Test Trap Tool** that is part of the OpenScape FM package. The tool can also be used to check the configured reactions of the OpenScape FM to certain traps (e.g. in the MAR environment).

Tool Installation:

Before the tool can be used, it must first be set up on the local computer.

This is done via a download that can be initiated via the Debug Options page (main menu entry **Server->Administration->Debug Options**) using the button **Download SNMP Trap Tool**.

The button opens a file browser in which the location and **File Name** of the ZIP archive containing the Test Trap Tool can be specified. The **Save** button triggers the download.

After a successful download the ZIP archive has to be unpacked.

Tool Execution:

If the unzipped directory of the tool is available, the tool can be started by calling `snmpTrapTool.bat` (Windows) or `snmpTrapTool.sh` (Linux) from within the directory.

The start of the tool opens a window in which a **Trap Type** can be selected, configured in **Trap Configuration** and finally sent with **Send Trap**.

For all traps that are to be sent, a source and a destination have to be specified.

The source (**From**) is the IP node for which a problem is to be created. And the destination (**To**) is the OpenScape FM server that should evaluate the trap together with the **Port** on which the OpenScape FM server expects the traps (default: `162`).

Three basic trap types can be selected:

- **Link Down** and **Link Down SPAM**: This is a generic link-down trap for the specified source. If the SPAM variant is selected, it is also possible to select how often (**Amount**) the trap should be sent in quick succession.
- **Link Up** and **Link Up SPAM**: This is the Link Up Trap matching to the first selection.
- **OS4K**: This generates a *VIP PHONE failure* alarm trap. The **Priority** determines the status of the alarm. `1` stands for *Minor*, `2` for *Critical* and `3` for *Warning*. The **Status** determines whether the alarm is on (`0`) or off (`1`).

The button **Send Trap** sends the configured trap to the selected target and port.

The button **Exit** closes the tool.

24 Database Files

Permanent information is stored in database files in the directory `<Default Install Dir>\server\database` (Windows) or `<Default Install Dir>/server/database` (Unix). There are three files:

```
server.db.base
server.db.base.old
server.db.base.delta
```

OpenScape FM stores any changes made to the system in the RAM and then writes these changes periodically to the `server.db.base.delta` file. When the size of the delta file reaches a certain limit, the entire RAM information is written to the `server.db.base` file.

Important Note:
Before performing an external backup of these database files, the OpenScape FM server has to be stopped! In order to save the files while the server is running, use the OpenScape FM specific backup mechanism, See *Chapter 19, "Backup and Restore"*

24.1 Resetting the Database

In order to delete all information stored in the OpenScape FM database the file `deleteFmDatabase` (Windows) or `deleteFmDatabase.sh` (Unix) to be found in the OpenScape FM installation directory has to be executed. The executable will stop the OpenScape FM server, erase the database and start the server again. Afterwards the passwort of root has to set again.

25 NAT Environment

The communication between the OpenScape FM server and the OpenScape FM client is achieved by the usage of the "Remote Method Invocation" (RMI) protocol. RMI allows the invocation of methods on other systems. For example the client will use an RMI call to perform an operation which is bound to a specific menu item (e.g. IP node: **IP->Status Poll**). To perform such a call the client requests a "remote reference" of the object on which a RMI call should be performed. This "remote reference" contains the IP address of the server on which the object is located. In case of NAT this will not be the IP address by which the server is visible to the client. The necessary mapping has to be done by the client in order to use the IP address by which the server is visible to the client.

The OpenScape FM server and client use a special mechanism to recognize a NAT environment. If needed the client will map the IP address to ensure a working RMI communication.

The same mechanism is also used to solve a similar problem by a OpenScape FM server running on a multihomed system.

If the OpenScape FM client has determined that it has to map the IP address of the "remote reference" this will be indicated. In the client GUI in the upper left corner, the toolbar will be extended by additional information:

- **NAT:** The toolbar will be extended by the term "NAT" followed by the real IP address.
- **Multihomed host:** The toolbar will be extended by the term "MHH" followed by the real IP address used for the RMI connection.

26 HTTPS and Certificates

26.1 What is HTTPS?

The HTTPS (HyperText Transfer Protocol Secure) is a technique to encipher and authenticate communication between web servers and web browsers within the World Wide Web. Technically HTTPS defines a new layer between HTTP and TCP.

The communication between web servers and web browsers uses the SSL protocol described in *Chapter 27, "SSL Encryption"*.

26.2 HTTPS within OpenScape FM

Within the OpenScape FM the HTTPS communication will be used by default between the OpenScape clients and the OpenScape FM server. Optionally the communication can be changed to HTTP mode.

Activation and deactivation of the HTTPS mode is done through a window within the OpenScape FM client. This window is opened by selecting the entry **Server->Administration->Server Properties...** from the main menu bar. The HTTPS configuration is located on the page **SSL Certificate**.

The page **General** is used to define which type of communication should be used:

- If the check box **Activate Https** is marked, the internal HTTP server of the OpenScape FM will be switched to HTTPS mode.
- If the check box **Activate RMI Over SSL** is marked, the RMI communication will also use SSL sockets (see *Chapter 27, "SSL Encryption"*).

If the connection mode gets reconfigured, the OpenScape FM server will be restarted automatically.

If an HTTP connection has been configured, the web clients can be started as before. Only the beginning of the connection URL has to be changed from `https://` to `http://`.

The OpenScape FM uses certificates for its secure communication for four different tasks:

1. For the WebServer communication with the ä Landing Page using Port `3043`.
2. For the RMI communication between Server and Client (also using Port `3043`).
3. For the WebServer communication with the Web Client and with the Performance Management Client using Port `3080`.
4. For the communication with IP nodes/objects within the OpenScape FM.

For the first three tasks by default the OpenScape FM uses self-signed certificates. This means the certificates necessary for the usage of HTTPS are generated by the OpenScape FM server itself. This has the advantage that certificates are instantly available after a reconfiguration and not only after the confirmation by an external certificate authority. Fees attached to this process are also avoided.

But there is also a disadvantage. The web browser will not acknowledge the provider of the certificate as trustworthy.

Note:
When the starting page of the OpenScape FM is loaded within a web browser using HTTPS, a warning will be stated by the browser. The browser will display that the security certificate was **NOT** provided by a trustworthy certificate authority.

How user defined certificates can be used instead, will be described in the following chapter (see *Section 26.3*).

If for a connection a self-signed certificate gets replaced by a user defined certificate, this ends the self-signing-mechanism for this connection.

For the fourth task, communication with IP nodes, certificates provided by the respective IP node are used. Usually these are automatically determined by the OpenScape FM during a Status Poll (see *IP Manager User Guide*). However, the certificates can also be loaded directly via the client interface (see *Section 27.3*).

26.3 Creation and Installation of a Customer Specific Certificate

If the internal web server of the OpenScape FM is started in HTTPS mode, the required private/public key pair and a self-signed certificate are generated by the server.

This self-signed certificate of the internal web server can be replaced by a customer specific certificate, which was signed by a Certification Authority (CA). Besides the encryption of the connection, this improves the security since the authenticity of the web server in ensured from the beginning, and the user has not to ignore browser warnings.

Important Note:
If a self-signed certificate has been replaced by a user defined certificate, an this runs out of time, it will *not* be automatically replaced/prolonged by a new self signed certificate.

Requesting, creating and importing certificates is supported by the OpenScape FM user interface (see *Section 26.3.1*). The integration of the certificates can also be performed manually (see *Section 26.3.2*).

The list of all certificates known to the OpenScape FM can be viewed as described in *Section 27.3*.

26.3.1 Creation of Certificates Using the User Interface

The configuration windows for editing certificates can be opened via the main menu entry **Server->Administration->Server Properties**. They are located on the page **SSL Certificate**.

On this page on the subpage **Port 3043** the certificate for the WebServer communication to the Landing Page and for the RMI communication between Server and Client can be configured.

On the subpage **Port 3080** the certificate for the WebServer communication to the Web Client and to the Performance Management Client can be set up with an identical procedure.

The data required to create the certificates is entered in the upper area of the configuration window.

The data necessary to generate a self-signed certificate can be entered within the same configuration window (The string enclosed in the brackets corresponds to the Distinguished Names (DN) within the certificate):

- The fields **Fully Qualified Domain Name (CN)** and **Subject Alternative Name (SAN)** are used to enter the host names for which the certificate should be generated.
- The fields **Organization (O)**, **Organization Unit (OU)**, **Email Address (EMAIL)**, **Country (C)**, **Locality (L)** and **State (ST)** define for whom the certificate should be generated.
- The certificate will be stored on the server within a keystore. This keystore will be protected by a password which can be chosen within the field **Password**.
- Within the field **Valid (days)** the validity period for the certificate can be defined. A new certificate will automatically be created by the server when the validity period is about to end.

Important Note:
In the Java Client, the same password is always used for the keystore and for the key. Therefore, the keystore password and key password must always be identical.

In the lower area, based on this data, the respective **Action** can then be triggered.

Among other things, a self-signed certificate can be created here, the data for the signing by a certification authority can be generated, or the response of the certification authority can be added to the keystore.

26.3.2 Manual Handling of the Certificates

In order to manually import a certificate, a custom password for the server-generated key has to be set initially. The password can be changed by selecting the entry **Server->Administration->Server Properties** from the main menu bar and using the page **SSL Certificate**. Here the subpages **Port 3043** (for the WebServer communication to the Landing Page and for the RMI communication between Server and Client) or **Port 3080** (for the WebServer communication to the Web Client and to the Performance Management Client) have to be used.

The private/public key pair generated by the OpenScape FM Server for **Port 3043** is stored using the alias `internalwebserver` in the file
`<inst_dir>/server/database/trustedcerts.jks`
where `<inst_dir>` is the OpenScape FM installation directory.

For **Port 3080** the alias `internaltomcatserver` in the file
`<inst_dir>/Wildfly/standalone/configuration/webappKeystore`
is used.

These files are a *Java Key Store (JKS)*, which is used to manage and protect public and private keys and the associated certificates.

To replace the self-signed certificate by a customer specific certificate, the following steps have to be performed for **Port 3043**:

1. Changing into the OpenScape FM installation directory and create a "Certificate Signing Request" for the generated public key by executing the following command. The program `keytool` is part of the *Java Runtime Environment (JRE)*.

   ```
   keytool -certreq -alias internalwebserver -keystore keystore.jks -file csr.req
   ```

On executing the command, the password assigned for the keystore will be requested. The key password and the store password must be the same.

2. The command generates a file named `csr.req`, which can be used to create a new certificate. The file must be sent to the appropriate *Certification Authority (CA)*.
3. After the certificate is received from the CA (e.g. `signed.cer`) the following command can be used to import it into the keystore. The OpenScape FM Server has to be stopped for the import. After the import the Server must be restarted manually.

   ```
   keytool -import -alias internalwebserver -keystore keystore.jks -file
   signed.cer
   ```

For **Port 3080** the alias has to be replaced with `internaltomcatserver`.

The following error might occur during the import of the certificate:

```
keytool error: java.lang.Exception: Failed to establish chain from reply
```

In this case the complete certificate chain has to be imported into the keystore. For this the same command must be used for all certificates of the chain, beginning with the root certificate of the CA.

```
keytool -import -alias root -keystore keystore.jks -file root.cer
```

Note:
If the generated private/public key pair should be replaced besides the certificate, the new key has to use RSA encryption and has to be stored in the keystore with the alias `internalwebserver` or `internaltomcatserver` respectively.

More about the java keytool can be found under:

```
http://download.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html
```

27 SSL Encryption

27.1 What is SSL?

When data is transferred across a network, someone who is not the intended recipient might get access to it and maybe modify it. Thus, steps must be taken to prevent unauthorized access and to ensure that the data has not been altered during transport.

On the internet, data encryption is often performed using the secure socket layer (SSL) protocol. It was originally designed by Netscape but can be used by most web browsers today. SSL provides two operations based on secure TCP sockets: the secret key exchange and the data encryption with the exchanged key.

An SSL connection between client and server is opened by the so called “SSL handshake”. During the initial operation, the server sends its public key to the client. The client then uses this public key to encrypt a generated symmetrical session key. This is a so called asymmetric encryption mechanism and it takes some time to be performed. For this reason, the first building of a connection will take longer than the subsequent data transfer (compare *Section 27.2, “Encryption”*).

Now, server and client both own a key and can use it for the data transfer: this symmetric encryption does not need more time than regular connections.

SSL works on TCP connections, thus it is an application level protocol which is inserted between the TCP layer and the application data protocol (see *Figure 39*).

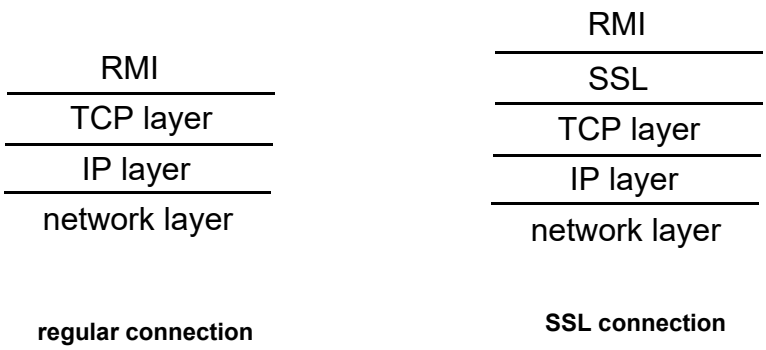


Figure 39 protocols needed in server-client data transfer

Thus an SSL connection is a secure, encrypted connection the data of which can not be read from systems other than the specific SSL server and client which have opened it.

27.2 Encryption

By default the OpenScape FM Server runs in SSL mode.

This can be modified by using the Client interface (see *Section 26.2, “HTTPS within OpenScape FM”*).

27.3 SSL Certificates

The SSL certificates can be loaded and handled within the OpenScape FM by using the main menu **Server->Administration->SSL Certificates**.

Handling Known Certificates:

Using the menu item **Show Certificates** displays a window which consists of three tabs that handle the SSL certificates currently known by the OpenScape FM:

- The tab **Untrusted Certificates** shows the certificates that are currently known but untrusted. The buttons below can be used to **Delete** a selected certificate from the OpenScape FM or to transfer it into the **Trust**ed mode.
- The tab **Trusted Certificates** shows the certificates that are currently known as trusted. The buttons below can be used to **Delete** a selected certificate from the OpenScape FM or to transfer it into the **Untrust**ed mode.
- The tab **Expiration Time** can be used to configure the creation of events that should be generated if the validity of a trusted certificate is about to end.

  Each line within the table of the tab represents one rule to create such events.

  New rules can be created, existing ones can be deleted, copied and the order of the rules within the table can be changed by selecting table entries and using the buttons below the table. The order of the rules has *no* effect on their execution.

  The **Network** and **Mask** of a rule define the IP range for which the rule is valid. If these values are left empty, the rule is valid for all IP nodes. Certificates will be tested against the IP of the host to which they belong. Generated expiry events will also be assigned to this IP node.

  The **Status** defines the status of the event that will be created as an expiration warning.

  The **Expiration Time** defines how much in advance of the certificate‘s end of validity the event will be created.

Loading New Certificates:

The menu item **Import Cert. from Server** can be used to import additional SSL certificates. These will be loaded from a certificate server that is identified by its host and its port number. The loaded certificates will be added as trusted certificates.

28 Mobile Access

Mobile Access allows the access to vital management information within the OpenScape FM by using an iOS or Android phone. This feature is very useful for service personnel or on-call duty by providing easy access to the OpenScape FM without the need to be in the office.

Mobile Access allows the accessing of events and the browsing of IP nodes from the OpenScape FM.

28.1 Technical Structure

The Mobile Client is implemented based on the architecture shown in *Figure 40*.

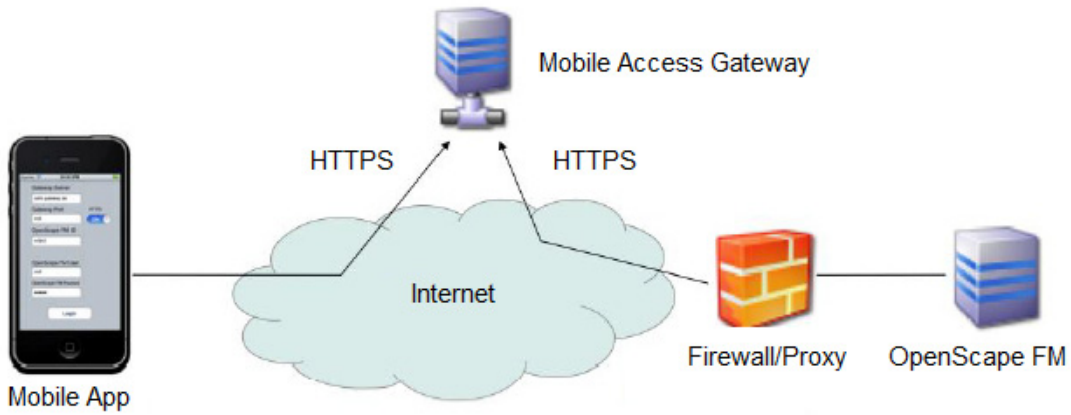


Figure 40 Technical Structure

The Mobile Client for iOS or Android has to be installed (see *Section 28.4.1*) on the phone as an App. The phone app establishes a connection to the desired OpenScape FM Server by using a Mobile Access Gateway. On the OpenScape FM Server an installed Mobile Access Server plugin will also communicate with the Mobile Access Gateway.

This means that all communications are directed forward to the Mobile Access Gateway, which allows the usage of very simple firewall rules.

In the Mobile Access Gateway both connections are met. Connections started by the client will be handed over to the connection started by the Mobile Access Gateway plugin of the OpenScape FM Server.

If the OpenScape FM Server can be directly reached by the App, no Mobile Access Gateway is needed

28.2 Mobile Access Gateway

The Mobile Access Gateway is responsible for passing the requests received by the Mobile Client to the corresponding OpenScape FM Server. It is also responsible for receiving the result data and forwarding it back to the correct client. When an OpenScape FM connects to the Mobile Access Gateway for the first time, it receives a unique ID. The Mobile Client will use this ID in order to define the OpenScape FM Server to which it wants to connect.

The Mobile Access Gateway is only needed if the OpenScape FM Server cannot be accessed directly via network by the Mobile Client

28.2.1 Mobile Access Gateway Installation

The Mobile Access Gateway has to be installed separately by starting the file `setup_mobilegw_osfm.exe` for Windows systems or `setup_mobilegw_osfm.sh` for Linux systems from the installation media.

The Gateway will install itself as a service called '`OpenScape FM Mobile Access Gateway.`'

The Mobile Access Gateway has to be installed on a system which is reachable via HTTP/HTTPS by both, the Mobile Access Client and the OpenScape FM Server.

An Oracle Java has to be installed on the system.

28.2.2 Mobile Access Gateway Configuration

The Mobile Access Gateway can be configured by editing the file '`rserver.properties`' which can be found in the '`<installation_dir>/conf`' directory.

The parameters '`httpport`' and '`httpsport`' in the properties file are used to define the ports used for HTTP and HTTPS connections (e.g. `httpsport=443`). These ports are used to receive requests from the Mobile Client and the Mobile Access Gateway plugin. If one of this parameters is missing, the respective service will be disabled.

The parameter '`keystore`' defines the path to the java key store (e.g. `keystore=conf/rsrv_keystore`). If the key store does not exist, it will be created during the next start of the gateway.

The parameter '`keypasswd`' defines the password to access the key store.

The parameter '`alias`' defines the certificate that should be used for HTTPS access. If the certificate does not exist, a self-signed certificate with this name will be created.

If HTTPS access is used, the same Java version has to be used for the Mobile Access Gateway and the OpenScape FM Server.

28.3 Mobile Access Gateway Plugin

The OpenScape FM has to provide its data to the web client. This task is achieved by the Mobile Access Gateway plugin.

The Mobile Access Gateway plugin communicates with the Mobile Access Gateway. It retrieves the requests send by the Mobile Client to the Mobile Access Gateway and processes them. Then it sends the results back to the Mobile Access Gateway, which returns them to the Mobile Client.

When the OpenScape FM Server connects to the Mobile Access Gateway it will register itself with an ID. This ID will be used by the Mobile Client in order to establish a connection to the OpenScape FM.

The ID and the general status of the Mobile Access Gateway can be displayed within the OpenScape by using the main menu entry **Server->Administration->Mobile Access->Status** within an OpenScape FM client.

28.3.1 Mobile Access Gateway Plugin Installation

The Mobile Access Gateway plugin is a plugin for the OpenScape FM. It will be installed during the installation process of the OpenScape FM.

To use it, it has to be initialized by selecting the main menu entry **Server->Plugins->Initialize Mobile Access plugin**.

28.3.2 Mobile Access Gateway Plugin Configuration

When the Mobile Access Gateway Plugin has been activated it can be connected to a Mobile Access Gateway.

This can be done be selecting the main menu entry **Server->Administration->Mobile Access->Configure...** within an OpenScape FM Client.

This action will open a window in which the connection data and the connection type to the Mobile Access Gateway can be entered. When a connection has been successfully established, the connection ID will be displayed in the field **Server ID**. This ID is needed in the Mobile Client to identify the OpenScape FM.

The button **Activate** / **Deactivate** can be used to activate or deactivate the connection for Mobile Clients.

Since the **Server ID** identifies an OpenScape FM, it will not be changed when the connection is deactivated/ activated. But the generation of a new ID can be enforced by using the **Clear** button. This will invalidate the old ID and a new one will be created during the next successful connection.

28.3.3 Personal View

Besides events and IP nodes every other object displayed in the OpenScape FM can be displayed in the Mobile Client. The objects that should be exported have to be configured in the OpenScape FM.

When an object search is performed in the Mobile Client, and the search option **Personal View** is used for the first time, a container object with the label **Mobile Access** is created in the submap of the user symbol of the related user. Objects that are copied into this container object can be viewed by the Mobile Client.

28.4 Mobile Client

The Mobile Client is an app that runs on an Apple or Android mobile phone. It provides the user interface to connect to an OpenScape FM Server and to display events and IP nodes.

After the Mobile Client is started, it connects to the configured Mobile Access Gateway and performs a login on the selected OpenScape FM Server. When the login has been performed, it will retrieve data by issuing further requests to the Mobile Access Gateway.

It is freely configurable which Mobile Access Gateway and which OpenScape FM should be used by the app.

28.4.1 Mobile Client Installation

The Mobile Client for iOS has to be loaded and installed from Apples app store.

The Mobile Client for Android has to be loaded and installed from the Google Play Store.

28.4.2 General Functions

The Mobile Client displays objects and events that are found on the respective OpenScape FM Server.

Generally the objects and events are displayed as lists in which each entry represents one object or event.

If list entries are tapped, more details about the entry will be displayed. If an arrow button on the right side is tapped, data related to the displayed entry will be displayed.

Individual **Tabs** are opened by using the buttons located at the bottom (iOS version) or by menu entries that can be swiped in from the left hand side (Android version).

The button displaying a circular arrow in the upper right corner can be used to reload the displayed data.

A button in the upper left corner displaying a list name can be used to go back to a previously opened list.

The following chapters contain a more detailed description the Mobile Client’s user interface (the screenshots are taken from the iOS version).

28.4.3 Login

When the Mobile Client app is started, it will ask for a login to a Mobile Access Gateway running anywhere or to an OpenScape FM Server running in a locally reachable net.

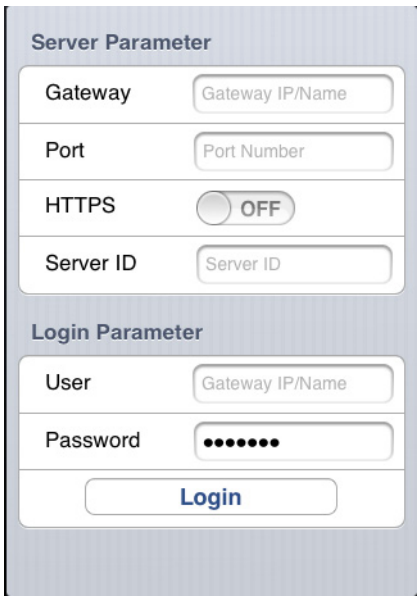


Figure 41 Login

The login window (see *Figure 41*) asks for the connection data and method. If a connection to a Mobile Access Gateway should be established, the **Server ID** of the desired OpenScape FM has to be entered. The **Server ID** can be displayed in the OpenScape FM Client by using the main menu entries **Server->Administration->Mobile Access->Status** or **Server->Administration->Mobile Access->Configure**.

The **User** and respective **Password** have to match a user in the OpenScape FM.

28.4.4 Overview

The **Overview** tab of the Mobile Client can be used to get a first impression of the status of the monitored objects on the OpenScape FM Server.

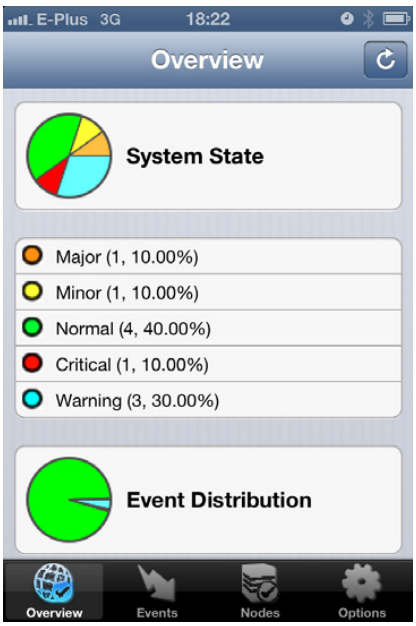


Figure 42 Overview

The tab (see *Figure 42*) provides an overview about the distribution of the system status of the monitored objects and the event status of the currently unacknowledged events. This enables a technician to see at a single glance whether their are problems that need attention.

If a status entry gets touched, a list will be displayed that contains all unacknowledged and uncorrelated events of the selected severity.

28.4.5 Events

The **Events** tab of the Mobile Client allows the access to the events that are shown in the Event Browser of the OpenScape FM. The list is ordered by date. The latest event is displayed on top of the list.

28.4.5.1 Event Search

By default the **Events** tab displays all events that are shown in the OpenScape FM Event Browser in 'Correlated View' (related events are merged into a single line).

To find specific events, the following parameters can be selected to reduce the displayed list to only those events that match all selected parameters (see *Figure 43*).

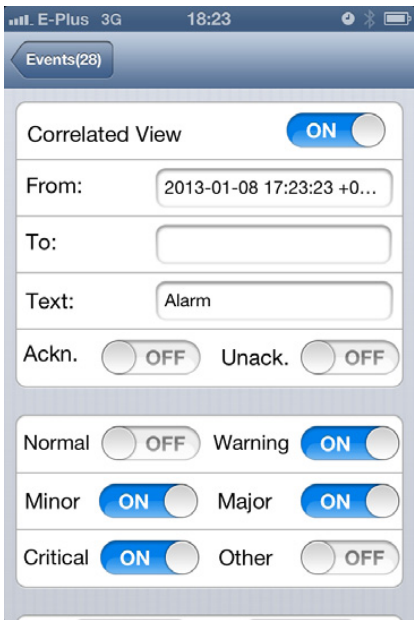


Figure 43 Event Search

- **From Date**, **To Date**: Only the events that happened within the time interval will be displayed.
- **Text**: The entered string must be a part of the **Source**, the **Category** or the **Description** of the event.
- **Severities**: The event must have one of the selected Severities.

28.4.5.2 Event Browser

The **Event Browser** (see *Figure 44*) lists the events that match a specified search criteria (see *Section 28.4.5.1, "Event Search"*) or all events if nothing was specified. The **Search** button on the upper left corner invokes the event search view.

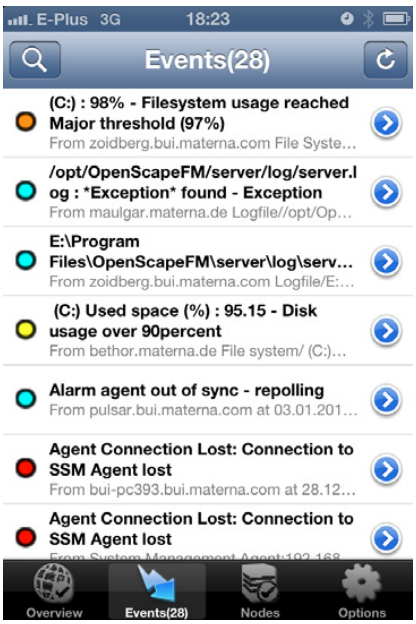


Figure 44 Event Browser

Tapping an event will open the Event Detail view (see *Section 28.4.5.3, "Event Details"*).

When correlated events exist for an event, an arrow button will be displayed on the right side of the event. Clicking this button will open a new event list, that only includes the correlated events.

28.4.5.3 Event Details

The Event Details dialogue opens when an event entry is tapped within the Event Browser. It shows detailed information about the selected event. The Acknowledge button can be used to acknowledge the event within the OpenScape FM. Correlated events will also be acknowledged.

28.4.6 IP Node and Object Browser

The tab **Nodes** provides access on IP nodes and objects that are managed by the OpenScape FM Server. It contains lists of IP nodes or of objects that reside on the 'Mobile Access' submap of the user, and it allows the navigation through the object tree.

28.4.6.1 Node Search

By default the **Nodes** tab displays all IP nodes that are shown in the OpenScape FM. The nodes are ordered by their status with the most critical nodes at the top of the list.

To find specific nodes, the following parameters can be selected to reduce the displayed list to only those nodes that match all selected parameters.

- **Text**: The entered string must be a part of the **IP Node Label**, the **Hostname** or the **IP Address** of the node.
- **Severities**: The node must have one of the selected Severities.

Within the Node Search the view can be changed to ‚**Personal View**'. In this case the browser will display a list of the objects that were chosen in the plugin (see *Section 28.4.6.2, "Node Browser"*), which are the objects that have been copied into the 'Mobile Access' container at the user's submap within the OpenScape FM.

Hierarchical Search can only be enabled when 'Personal View' is selected. If enabled, a 'deep search' is performed in the object trees below the 'Mobile Access' container. If it is disabled, only the direct child objects of the 'Mobile Access' container are searched.

28.4.6.2 Node Browser

The **Node Browser** (see *Figure 45*) lists the IP nodes that match a specified search criteria (see *Section 28.4.6.1, "Node Search"*) or all nodes if nothing was specified. The **Search** button on the upper left corner invokes the node search view.

Figure 45 Node Browser

Tapping a node will open the Object Detail view for the current IP node (see *Section 28.4.6.4, "Object Details"*).

When the arrow button on the right side of the node is pressed, an object list will be opened (see *Section 28.4.6.3, "Object Browser"*). This list contains the objects that are located on the submap of the IP node. This feature makes it possible to navigate through the object tree of the OpenScape FM.

28.4.6.3 Object Browser

The **Object Browser** lists the objects of a selected submap.

If a listed object has child objects, the arrow button on the right side of the object can be pressed to open an Object Browser. The browser contains the child objects of the current object.

The button **Back** in the upper left corner opens the previous view (generally a list with the objects from the submap of the parent object).

This last two functions allows the navigation through the object tree of OpenScape FM.

Tapping an object will open the Object Detail Browser (see *Section 28.4.6.4, "Object Details"*) for the selected object.

28.4.6.4 Object Details

The Object Details dialogue opens when an IP node entry is tapped within the Node Browser or an object is tapped in the Object Browser. It shows detailed information about the selected object. The information may be different depending on the type of the selected object.

28.4.7 Options

The **Options** tab can be used to logout or to display or delete the stored HTTPS certificates.

29 String Formatting Language

The String Formatting Language can be used within various plugins to flexibly create output based on any number of input variables.

This chapter describes the structure of this language in detail.

29.1 The Language in BNF Notation:

29.1.1 Tokens:

```
<DEFAULT> TOKEN: {
<ESCAPE: "\\" ["{","}","$",",","\\","n","t"]>
| <NESCAPE: "\\" ~["{","}","$",",","\\","n","t"]>
| <TEXT: (~["{","}","$",",","\\"])+>
}
```

29.1.2 NON-Terminals

Table 6 NON-Terminals

Input	:==	StartPhrase <EOF>
Escape	:==	(<ESCAPE>)
Text	:==	(<TEXT> | <NESCAPE>)
Word	:==	Escape | Text
Get	:==	(("$get{" \| "${") Phrase "}")
GSet	:==	"$gset{" Phrase "," Phrase "}"
Set	:==	"$set{" Phrase ("," Phrase |) "}"
If	:==	"$if{" Phrase "," Phrase ("," Phrase)? "}"
Switch	:==	"$switch{" Phrase ("," ("{" Phrase "," Phrase "}"))* "}"
Match	:==	"$match{" Phrase "," Phrase "," Phrase "}"
Split	:==	"$split{" Phrase "," Phrase "," Phrase "}"
Length	:==	"$length{" Phrase "}"
SubString	:==	"$substring{" Phrase "," Phrase ("," Phrase)? "}"
IndexOf	:==	"$indexof{" Phrase "," Phrase ("," Phrase)? "}"
LastIndexOf	:==	"$lastindexof{" Phrase "," Phrase ("," Phrase)? "}"
ReplaceFirst	:==	"$replacefirst{" Phrase "," Phrase "," Phrase "}"
ReplaceAll	:==	"$replaceall{" Phrase "," Phrase "," Phrase "}"
Range	:==	"$range{" Phrase "," Phrase ("," Phrase)? "}"
Array	:==	"$array{" Phrase ("," Phrase)* "}"
Add	:==	("$+{" \| "$add{") Phrase "," Phrase "}"
Sub	:==	("$-{" \| "$sub{") Phrase "," Phrase "}"
Mul	:==	("$*{" \| "$mul{") Phrase "," Phrase "}"
Div	:==	("$/{" \| "$div{") Phrase "," Phrase "}"
Mod	:==	("$%{" \| "$mod{") Phrase "," Phrase "}"
Math	:==	(Add | Sub | Mul | Div | Mod)

Table 6 NON-Terminals

Bool	:==	`( Or | And | Matches | NotMatches | Equals | NotEquals | Less | LessEquals | Greater | GreaterEquals | Not )`
Matches	:==	`( "$~{" ) Phrase "," Phrase "}"`
NotMatches	:==	`( "$!~{" ) Phrase "," Phrase "}"`
Or	:==	`( "$||{" ) Phrase "," Phrase "}"`
And	:==	`( "$&&{" ) Phrase "," Phrase "}"`
Equals	:==	`( "$=={" ) Phrase "," Phrase "}"`
NotEquals	:==	`( "$!={" ) Phrase "," Phrase "}"`
Less	:==	`( "$<{" ) Phrase "," Phrase "}"`
LessEquals	:==	`( "$<={" ) Phrase "," Phrase "}"`
Greater	:==	`( "$>{" ) Phrase "," Phrase "}"`
GreaterEquals	:==	`( "$>={" ) Phrase "," Phrase "}"`
Not	:==	`( "$!{" ) Phrase "}"`
FormatDate	:==	`"$formatdate{" Phrase "," Phrase ( "," Phrase ( "," Phrase)? )? "}"`
Parsedate	:==	`"$parsedate{" Phrase "," Phrase ( "," Phrase ( "," Phrase )? )? "}"`
LogError	:==	`"$_logError{" Phrase ( "," Phrase )* "}"`
LogWarn	:==	`"$_logWarn{" Phrase ( "," Phrase )* "}"`
LogInfo	:==	`"$_logInfo{" Phrase ( "," Phrase )* "}"`
StartPhrase	:==	`( Word | Get | Set | GSet | Script | Switch | If | Bool | Match | Split | Length | IndexOf | LastIndexOf | ReplaceFirst | ReplaceAll | Range | Array | SubString | Math | FormatDate | ParseDate | "," | "$" )*`
Phrase	:==	`( Word | Get | Set | GSet | Script | Switch | If | Bool | Match | Split | Length | IndexOf | LastIndexOf | ReplaceFirst | ReplaceAll | Range | Array | SubString | Math | FormatDate | ParseDate | "{" Phrase "}" )+`

29.2 The Functionality of the Different Statements

29.2.1 The Get Statement

`$get{ Phrase`$_{in}$` } or ${ Phrase`$_{in}$` }`

Returns the value of the variable with a name matching the return value of `Phase`$_{in}$.

29.2.2 The GSet Statement

$gset{ $Phrase_{var}$, $Phrase_{value}$ }

The GSet Statement will globally set the value of the variable with the name defined by $Phrase_{var}$ to the result generated by $Phrase_{value}$.

29.2.3 The Set Statement

$set{ $Phrase_{var}$, $Phrase_{value}$ }

The Set Statement in the form shown above will work like the GSet Statement, but the variable will only be defined locally to be used in the enclosed sub statements within the enclosing block ({}).

The statement may also be used omitting the second phrase:

$set{ $Phrase_{var}$ }

In this case the variable with the name defined by $Phrase_{var}$ will not be set to a value but removed instead.

29.2.4 The Switch Statement

$switch{ $Phrase_{in}$ <,{ $Phrase_{1A}$, $Phrase_{1B}$ } >
<, { $Phrase_{2A}$, $Phrase_{2B}$ } >
...
< , { $Phrase_{NA}$, $Phrase_{NB}$ } > }

The Switch statement will return the $Phrase_{xB}$ when the input $Phrase_{in}$ matches the regular expression $Phrase_{xA}$ (x being between 1 and N).

For example the Phrase

```
$switch{${severity},{Warning,Medium},{Minor,Medium},
{Major,Medium},{Critical,High},{.*,Low}}
```

provides the result ’Medium’ when the variable ’severity’ has the value ’Warning’, ’Minor’ or ’Major’. It provides the value ’High’ when the variable ’severity’ has the value ’Critical’. And it provides the value ’Low’ in all other cases.

29.2.5 The Match Statement

$match{ $Phrase_{in}$, $Phrase_{expr}$, $Phrase_{array}$ }

This statement will parse the result of $Phrase_{in}$ with a Perl like regular expression which results from $Phrase_{expr}$. The regular expression should contain any number of sub expressions encloses by round brackets. Each sub expression that can be successfully parsed will copy its content to a new member of an array with the name generated by $Phrase_{array}$.

E.g. the statement

`$match{ abcdefghijk , .*(c).*(f.*i).* , testarray }`

will generate the array 'testarray' which consists of the two entries 'c' and 'fghi'. To access the entry 'c' of the testarray you have to use the following `testarray[0]`.

29.2.6 The Split Statement

$split{ Phrase_in , Phrase_splitter , Phrase_array }

This statement will search the string generated by $Phrase_{in}$ for occurrences of the substring generated by $Phrase_{splitter}$. These occurrences will be removed from $Phrase_{in}$ and the different remaining segments will be placed within an array with a name generated by $Phrase_{array}$.

E.g. the statement

`$split{ abxcdxef , x , testarray }`

will generate the array 'testarray' which consists of the three entries 'ab', 'cd' and 'ef'. To access the entry 'ab' of the testarray you have to use the following `testarray[0]`.

29.2.7 The Length Statement

$length{ Phrase_in }

Returns the length in characters of the output of $Phrase_{in}$.

29.2.8 The Substring Statement

$substring{ Phrase_in , Phrase_start }

$substring{ Phrase_in , Phrase_start , Phrase_end }

This statement will return a part of the string generated by $Phrase_{in}$.

In the first case the returned part will be the rest of string beginning at position $Phrase_{start}$ (where $Phrase_{start}$ must result in an Integer value).

In the second case the returned part will be the substring beginning at position $Phrase_{start}$ and ending at position $Phrase_{end}$ (where $Phrase_{start}$ and $Phrase_{end}$ must result in Integer values).

29.2.9 The Indexof Statement

$indexof{ Phrase_in , Phrase_match }

$indexof{ Phrase_in , Phrase_match , Phrase_pos }

In the first case shown above the statement will return the position of the leftmost occurrence of the result of $Phrase_{match}$ within the result of $Phrase_{in}$.

In the second case the substring search begin at the position defined by $Phrase_{pos}$ (which must result in an Integer value). Possible matches which will occur left of this position will be ignored.

29.2.10 The Lastindexof Statement

$lastindexof{ Phrase_{in} , Phrase_{match} }

$lastindexof{ Phrase_{in} , Phrase_{match} , Phrase_{pos} }

The same as the Indexof Statement. But in this case the rightmost instead of the leftmost matching position will be returned.

29.2.11 The Replaceall Statement

$replaceall{ Phrase_{in} , Phrase_{old} , Phrase_{new} }

This statement will return the string created by $Phrase_{in}$ in which all occurrences of the substring created by $Phrase_{old}$ will be substituted by the string created by $Phrase_{new}$.

29.2.12 The Replacefirst Statement

$replacefirst{ Phrase_{in} , Phrase_{old} , Phrase_{new} }

The same as the Replaceall Statement, but only the first (leftmost) occurrence of $Phrase_{old}$ will be replaced by $Phrase_{new}$. Additional occurrences will be left unchanged.

29.2.13 The Range Statement

"$range{" Phrase_{StartValue} "," Phrase_{EndValue} ("," Phrase_{Increment})? "}"

The command provides a sequence of values by simulating a classical For loop.

The generated values start with $Phrase_{StartValue}$ and $Phrase_{Increment}$ is added until the next step would exceed $Phrase_{EndValue}$. If $Phrase_{Increment}$ is not defined it will be set to 1.

Important Note:
The statement does not check whether the definition will provide a very large or event infinite sequence of values.

Example:

$range{0,10,2}

The statement generates {0, 2, 4, 6, 8, 10}, which is the sequence of values from 0 to 10 in steps of 2.

29.2.14 The Array Statement

"$array{" Phrase$_1$ ("," Phrase$_n$)* "}"

The statement generates the listed values as a sequence. Any number of values can be listed.

Example:

$array{a,1,b,2}

Generates a sequence consisting of the values "a", "1", "b" and "2".

29.2.15 The Math Statement

(Add | Sub | Mul | Div | Mod)

The math statement returns the value of the mathematical operation.

29.2.15.1 Add Operation

The Add operation sums the two given parameter.

Example:

$+{2,3} or $add{2,3} return the value 5

29.2.15.2 Sub Operation

The Sub operation subtract the two given parameter.

Example:

$-{5,3} or $sub{5,3} return the value 2.

29.2.15.3 Mul Operation

The Mul operation multiplies the two given parameter.

Example:

$*{2,3} or $mul{2,3} return the value 6.

29.2.15.4 Div Operation

The Div operation divides the two given parameter.

Example:

`$/{6,3}` or `$mul{6,3}` return the value 2.

29.2.15.5 Mod Operation

The Mod operation returns the modulus of the division of the two given parameter.

Example:

`$%{7,3}` or `$mod{7,3}` return the value 1.

29.2.16 The If Statement

"$if{" $Phrase_{Bool}$ "," $Phrase_{if}$ ("," $Phrase_{else}$)? "}"

This statement can be used to make decisions. The first statement has to be an boolean statement. If its output is true, the second statement will be executed. If the output of the boolean statement is false, the third statement will be executed, when it is stated.

Example:

```
$if{$~{${severity},Critical},Important,Unimportant}
```

If the severity of an event is Critical, the output is „Important“. For the other severity values the output is „Unimportant“.

29.2.17 The Bool Statement

```
( Or | And | Matches | NotMatches | Equals | NotEquals | Less | LessEquals | Greater
| GreaterEquals | Not )
```

The Bool statements are used to define boolean operations. The following operations are possible.

29.2.17.1 Or Operation

("$||{") $Phrase_1$ "," $Phrase_2$ "}"

This operation returns true when one of the Phrases is true.

29.2.17.2 And Operation

("$&&{") $Phrase_1$ "," $Phrase_2$"}"

This operation returns true when both Phrases are true. The result is always false in other conditions.

29.2.17.3 Matches Operation

`( "$~{" )` $\text{Phrase}_{\text{string}}$ `","` $\text{Phrase}_{\text{reg}}$ `"}"`

This operator compares, if the first Phrase matches the given Regular expression $\text{Phrase}_{\text{reg}}$. Please keep in mind, that this operation is „greedy“. That means, the comparison will be done with the complete Phrases.

Note:
The Java class `java.util.regex.Pattern` is used to handle Regular Expressions. Since Regular Expressions are a very powerful instrument, an exhaustive explanation cannot be provided at that point. Please refer to the specific Java documentation about the class for detailed information.

Example:
`$~{"It is Critical","Critical"}` will return false.

`$~{"It is Critical",".*Critical.*"}` will return true.

29.2.17.4 NotMatches Operation

`( "$!~{" ) Phrase "," Phrase "}"`

This operator compares, if the first Phrase does not match the given Regular expression $\text{Phrase}_{\text{reg}}$. Please keep in mind, that this operation is „greedy“. That means, the comparison will be done with the complete Phrases.

Example:
`$!~{"It is Critical","Critical"}` will return true.

`$!~{"It is Critical",".*Critical.*"}` will return false.

29.2.17.5 Equals Operation

`( "$=={" ) Phrase "," Phrase "}"`

This operator compares, if the given Phrases are equal. When they are equal it returns true.

29.2.17.6 NotEquals Operation

`( "$!={" ) Phrase "," Phrase "}"`

This operator compares, if the given Phrases are not equal. When they are not equal it returns true.

29.2.17.7 Less Operation

`( "$<{" ) Phrase "," Phrase "}"`

This operator compares, if the first Phrase is less than the second one.

29.2.17.8 LessEquals Operation

```
( "$<={" ) Phrase "," Phrase "}"
```

This operator compares, if the first Phrase is less or equal than the second one.

29.2.18 The FormatDate Statement

```
$formatdate{<date in milli seconds>, <format of the output date>[, locale, [country]]
```

The formatdate statement returns a date. See the following table for format variants. .

Table 7 Date and Time Patterns within the Java Class SimpleDateFormat

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Examples:

Use `$formatdate{1258626787000,dd.MM.yyyy}` to get a date like this `19.11.2009`.

Use `$formatdate{1258626787000,dd. MMMM yyyy,en}` to get a date like this `19. November 2009`.

29.2.19 The ParseDate Statement

`$parsedate{{<date>, <format of the given date>[, locale, [country]]`

The parsedate statement parses a given date and returns the date as number of milli seconds since 1.1.1970. The format parameter describes the format of the given date. See Table 7 on page 182 for format options.

Examples:

Use `$parsedate{19.11.2009,dd.MM.yyyy}` to get a date like this `1258626787000`.

Use `$formatdate{19. November 2009,dd. MMMM yyyy,en}` to get a date like this `1258626787000 with another given date format`.

29.2.20 The LogError LogWarn and LogInfo Statements

$_logError{ $Phrase_1$ <, $Phrase_2$>
...
<, $Phrase_N$> }

$_logWarn{ $Phrase_1$ <, $Phrase_2$>
...
<, $Phrase_N$> }

$_logInfo{ $Phrase_1$ <, $Phrase_2$>
...
<, $Phrase_N$> }

The _logError, _logWarn and _logInfo statements can be used to add information to the `server.log` logfile. Depending on the used statement, the comma separated phrases will be included as an Error, Warning or Information.

A The Rights of the Base Module

The base module access rights are integrated into the general access management (*see Chapter 15, “Functions of the Rights”*).

The description of the individual rights can be found within the tooltipps for the corresponding right symbols (tree or submap).

The names of the rights for the base module begin with the designation *Base*.

B Prerequisite Hardware and Software Environment

The hardware and software requirements for OpenScape FM Server and Clients can be found within the Unify Release Nodes for OpenScape FM.

Virtualization:
It is allowed to install the OpenScape FM on a virtual system running on a VMware vCenter.

Important Note:
The OpenScape FM support does not cover problems which are caused by the interaction of the guest operating system and hypervisor or which can be assigned to one of them.
Other problems occurring in a constellation where the OpenScape FM is running on a virtual system on a VMware vCenter will be analyzed as usual as part of the normal support process.

C Installation Process

This part of the User Guide describes the installation of the OpenScape FM. It is presumed that additional software, like the operation system or database, is available and installed.

A detailed description of the installation can be found in the separate *Installation Guide* provided by Unify Software and Solutions GmbH & Co. KG.

During the installation one of three scopes can be selected for the installation:

- The full OpenScape FM installation is the default installation with all components.
- OpenScape Business / HiPath 3000 only installs the System Management and OpenScape Business / HiPath 3000 Plugins.

C.1 Using MySQL

When MySQL is used for OpenScape FM in connection with the Performance Management plugin, the following steps have to be considered:

1. **Database Driver Installation** (this part can be skipped if **MySQL 5.5** is used)

   The database driver (JDBC driver from MariaDB) which is included in the OpenScape FM installation is only supported for MySQL 5.5. If a different MySQL version is used, the corresponding MySQL driver has to be installed manually. This can be done as follows:

   a) Open the download page `http://dev.mysql.com/downloads`

   b) Select the link `MySQL Connectors`

   c) Select the link `Connector/J`

   d) At the combo box **Select Platform**, select `Platform Independent`

   e) Download the archive (available as `zip` or `tar`)

   f) Extract the archive file and locate the driver. Its name is something like `mysql-connector-java-X.X.XX-bin.jar` where `X.X.XX` represents the current version number

   g) Stop the OpenScape FM server process

   h) Copy the driver file `mysql-connector-java-X.X.XX-bin.jar` into the directory `<OPENSCAPE FM INSTALL>/server/lib/external`

   i) **Delete** the file `<OPENSCAPE FM INSTALL>/server/lib/external/mariadb-java-client.jar`. **This step is important**, otherwise the database connection may not work correctly.

   j) Start the OpenScape FM server process

2. **Preparation of the Database**

   The Performance Management Plugin needs a pre-defined database. To create a database in MySQL, perform the following steps:

a) Connect to the local MySQL database server by invoking the `mysql` command line tool. If the tool is not found, maybe the `mysql-client` package is not installed. The client prompts for a password for "`root`" if invoked as stated below. This is **not** the Unix-root user but the MySQL database administrator. The password has to be known. It is usually defined during the MySQL server installation process.

   Command: `mysql -u root -p`

b) Create a database with a name of your choice. The same name has to be configured in the OpenScape FM user interface using the main menu entry **Add-Ons->Performance Management->Select or Create Database Connection**.

   Command: `create database pm_database;`

c) Exit the command line client

   Command: `exit;`

C.2 Linux

Details about the requirements on the operating system and the JVM can be found in *Appendix B, "Prerequisite Hardware and Software Environment"*.

In order to install OpenScape FM under Linux , the script `openscapefm.bin` has to be started on the installation media:

```
sh /setup_osfm.sh
```

A command line interface will guide through the installation process, which is – concerning the steps which are performed – very similar to that on the Windows Client. Therefore, we will describe the installation on a Windows machine in detail.

The default installation directory on Linux systems is `/opt/OpenScapeFM` .

C.3 Windows

The installation will be triggered by starting the file `setup_osfm.exe` on the installation media. This will start an installer that will perform the installation.

First the installer looks for installed virtual machines for Java(TM) and will use the required version. If no virtual machine can be found, the installation of OpenScape FM will be canceled.

A welcome message will be displayed, the installation files will be extracted and the graphical installation interface will be displayed which is used to configure the installation. The button **Next** is always used to progress to the next step. The button **Cancel** can be used to abort the installation.

To start the configuration, first the License Agreement has to be accepted.

If this is done, it can be selected which components should be installed.

Either a full OpenScape FM installation or an OpenScape Business installation can be performed.

In the following window the installation directory can be defined. If this is the first installation, the default installation directory `C:\Program Files\OpenScape FM` will be displayed. If it is not the first installation, the directory of the last installation will be displayed. By pressing the button **Choose...** any user-defined directory can be selected.

During the installation the system/port on which the Customer License Agent is running has to be configured. If no product license has been provided, a demo license will be created during the first connection. This license activates the software for 90 days. Licenses are entered by using the mechanisms of the CLA. The menu entry **Server->Administration->License Manager->Load License File...** is only used to activate licenses for the HiPath QM and UM which are managed by the OpenScape FM.

After a confirmation of the installation directory the actual installation process starts.

If the OpenScape FM should monitor other hosts using WMI, an account for this function can be entered during the installation process. This account must belong to the local group *Administrator* and must have the Windows security policy *'Logon as a Service'*. For this account the **Domain**, **Service User** and **Password** have to be entered.

Within the window the progress of the installation will be displayed and, if successful, confirmed by a green check mark.

If the OpenScape FM Server or the Event Gateway have been installed, they will be automatically started as the service OpenScape FM Startup Service or Event Gateway.

A web browser or a stand alone client can now be used to connect to a Server by using the URL `https://localhost:3043`.

C.4 Installation Parameters

To modify the installation process, some parameters may be added to the installation command. These parameters are described within this paragraph and can be used for both versions of the installer: Linux and Windows.

- `-console`
  If this parameter is used, the installation will be run within a console window instead of within the standard installation user interface. This will offer the same configuration options as the standard installation.
- `-silent`
  If this parameter is used, the installation will be performed as a background process. Neither the standard installation interface nor a console window will be displayed. The installation uses the default installation directory defined for the respective operation system.
- `-Dnobackup=true`
  The installation will be performed without a backup of the existing OpenScape FM installation.
- `-Dcreatebackup=true`
  The installation will be performed including a backup of the existing OpenScape FM installation.
- `-Dserver=true`
  If the installation is performed as a background process, this parameter can be used to perform a full installation of the OpenScape FM.

- `-Dserver_osbiz=true`
  If the installation is performed as a background process, this parameter can be used to select the installation option "OpenScape Business".
- `-Dtarget.directory="<installation_dir>"`
  This parameter can be used to define an installation directory. The installation will be made into the directory defined by the installation path entered instead of `<installation_dir>`. If OpenScape FM is already installed on the system, this parameter will be ignored and the already existing installation directory will be used.

A possible command for Linux would be:

```
sh setup_osfm.sh -Dtarget.directory=/opt/OpenScapeFM_Test -silent
```

C.5 Deinstallation

- On a Linux machine, the following command can be used:
- `/<OpenScape installation directory>/uninstall.sh`
- On a Windows system, OpenScape FM can be removed by using the entry “Software” in the “Control Panel”. “OpenScape FM” has to be selected in the list of applications and the button “Remove/Deinstall” has to be pressed. Following the instructions in the dialogue will deinstall the fault management.

If a registered Tray Bar Symbol (see *Chapter 9, “Display of Tray Bar Icons”*) exists when the deinstallation is started, the symbol will not be deinstalled automatically. The deinstallation of the symbol has to be started manually by using the entry **Deinstallation...** from the context menu of the Tray Bar Symbol.

D Configuration of the System

D.1 Initialisation of Plugins

After a successful installation (see *Appendix C, "Installation Process"*) the OpenScape FM can be configured.

When a Client has been started (see *Section 4.2, "Starting the Client"*), the additionally needed plugins can be initialized. This can be done by selecting the respective entry from the **Server->Plugins** menu.

The plugins IP Manager, System Management, Layer 2 Manager, Control Center, Performance Management and Enterprise MIB will be automatically initialized during the installation process.

If the OpenScape Business version has been installed, only the plugins IP Manager, System Management and OpenScape Business will be available and these will be automatically initialized during the installation process.

D.2 Change the Java Version

OpenScape FM is implemented in java. In order to run the OpenScape FM server and the client a java runtime environment (JRE) or java development kit (JDK) has to be available on the system. The client will be started as a java applet which uses the JRE which is defined in the system properties. The OpenScape FM server uses the JRE/JDK which has been choosed at the installation. If this version will be deinstalled, at a new start of the OpenScape FM server, the newest JDK/JRE instance with a needed version will be searched and set automatically. If another version should be used by the OpenScape FM server, this version can configured manually. For windows the program `ServerProperties.exe` and for linux the shell script `serverProperties` can be used. This program can be found in the installation directory of the OpenScape FM.

The variable JAVA_HOME should point to the upmost directory of the JDK installation.

Windows:
A double click of the file `ServerProperties.exe` starts the program. A file browser appears which can be used to select the new JAVA-HOME directory. You can assign it by clicking the button **Choose**. Clicking the button **Refresh** the current directory will be refreshed. The button **Exit** will close the program and changes nothing in the settings.

Linux:
The command `serverProperties` opens a graphical Interface similar to the Windows program. If it is not possible to open this graphical Interface a command line interface will be started. Here the path to the JAVA HOME has to be entered manually and to be accepted by the Enter key.

After changing the path the Startup Service has to be restarted for windows and linux.

E Licensing

E.1 General

The usage of the OpenScape Desktop (DT) and its plugins is constrained by specific licenses representing different features. Each license enables the usage of a specific **feature** of the OpenScape Desktop and Fault Management.

After the OpenScape DT is installed an instant demo license for 90 days is generated during the first connection to the Customer License Agent. This licenses contain entries for the OpenScape DT and the covered Fault Management (FM) features. After the demo period has elapsed the Fault Management functions are automatically deactivated. Useful working with the OpenScape FM is no longer possible.

Therefore it is recommended that an OpenScape Desktop license will be installed.

OpenScape PlugIns integrate the different HiPath technologies (HiPath 3000/5000, HiPath 4000 etc.) into the OpenScape Desktop and Fault Management. The plugins can operate either in Desktop or in Fault Management mode. When the plugins are operating in Desktop mode all related systems of the technologies can be discovered and organized on the submaps, restricted system information can be queried and dedicated system functions can be called. In Fault Management mode additionally the systems will show a graphical fault state and a variety of technology specific fault information can be accessed.

The OpenScape Fault Management licensing is based on Ports for specific Technology Levels (see Appendix E.4, “Technology Levels and Ports”).

An OpenScape DT can contain several activated plugins, which are either operating in FM mode or in DT mode. If the FM-capable plugins shall operate in FM mode the existence of a specific license (HiPathFaultManagement feature) is required. The installation of this license will activate the FM functions of the plugins.

To activate OpenScape FM Desktop and IP Manager functions, both an OpenScape FM Desktop license and an IP Manager license have to be activated.

E.2 License Management

The License Management System (LM) developed by Unify GmbH & Co. KG represents a tool for managing licenses. To avoid concurrent licensing mechanisms the OpenScape FM is able to use the LM, which makes it possible to have only one integrated license management for the whole OpenScape environment. In the following it is assumed, that the reader has knowledge of the Licensing methods. More about this can be found in the respective documentation.

The LM allows the usage of floating licenses. Floating licenses are not bound to named systems. They are only limiting the number of OpenScape FM Servers running at the same time. This allows a more efficient use of fewer licenses by sharing them in the network. License Administrators can control who uses the licensed applications and the machines for which licenses should be available.

Floating licenses are only bound to the Customer License Agent (CLA) which provides the licenses, but not to the OpenScape FM server that is requesting a license.

E.2.1 Licensing Preconditions

A generated license must be initialized and active on the CLA. Otherwise the OpenScape FM cannot request any licenses.

Within this environment the OpenScape FM will work like a so called Customer License Client (CLC). The OpenScape FM will request OpenScape FM specific licenses (licensed features) from the CLS using the CLA. To allow the OpenScape FM Server to start, all requested OpenScape FM license features will be required.

The OpenScape FM server and the CLA can reside on the same system or on different systems.

E.2.2 License File and Configuration

The OpenScape FM uses a license file that contains the connection data to the Customer License Agent (CLA). The data is configured during the installation process and the file will be generated automatically.

The connection data to the CLA can be changed from within the OpenScape FM by selecting the main menu entry **Server->Administration->License Manager->Configure Customer License Agent (CLA)**, which will open a configuration window.

Within the configuration window, the following data can be configured:

CLA IP address/host:
The IP address or the hostname of the CLA.

CLA port:
The port that should be used to connect to the CLA. The default port is 61740.

Connection timeout (ms) and Number of connection retries:
When a request is send to the CLA is has to respond within the given interval. If not, the request will be retransmitted. If the max number of retries is reached without getting a response, an error representing a ’lost connection’ will be generated.

If the **OK** button is pressed, a license file containing the entered data will be created. This file will overwrite the existing OpenScape FM license file.

Important Note:
If an invalid connection is entered or no license is available on the CLA, the license can no longer be checked. In this case the OpenScape FM Server will be stopped.

E.2.3 Licensing Check

All license features are automatically checked once a day. Furthermore a user can manually perform a license check be interrogating the license information from the license manager. This can be done by selecting the entry **Server->Administration->License Manager->License Information** from the main menu.

If such a check is performed, the appropriate license features will be requested from the CLA. This includes the actual state (available, unavailable, pending), the remaining time and quantities.

The quantity of licenses for the requested features will always be 1 for one OpenScape Server. If greater quantities are licensed, this means that more than one OpenScape FM Server can request licenses from the same CLA at the same time.

License Errors will be mapped into the OpenScape FM and will be displayed within the Event Browser.

E.3 Installing a License File

For nearly all plugins and functions the licensing is handled by the CLA as described in the previous chapter.

Exceptions are the HiPath User Management (HPUM) and the HiPath Quality Management (HPQM). These are activated by an internal licensing mechanism for which a matching license file has to be loaded.

There are two mechanisms to install a license file. The first mechanism is via the OpenScape FM client (Appendix E.3.1, “Entering a License File via the Client”) and the second is via a standalone program (Appendix E.3.2, “Entering a License File via Standalone Program”).

E.3.1 Entering a License File via the Client

For this method a client has to be started first and a logon to the desktop has to be performed. Therefore the OpenScape FM server must be running. By selecting the menu item **Server->Administration->License Manager->Load License File** from the main menu, a file chooser will be started allowing to select a license file for installation.

Important Note:
If the license feature of the MSMC Desktop has been expired (server was stopped automatically) and a newlicense shall be integrated the standalone program has to be used, please refer to Section E.3.2, “Entering a License File via Standalone Program”.

Note:
Cannot be used to load CLA licenses.

E.3.2 Entering a License File via Standalone Program

In the installation directory of the MSMC FM contains the program `installLicenseFile.exe` (Windows) or `installLicenseFile` (Unix). This program allows to select a license file and performs the installation. A valid license file has to be installed when the MSMC DT server is not running and cannot be started because of a missing license.

Note:
Cannot be used to load CLA licenses.

E.3.3 Impact on Existing Licenses

A new license overwrites already existing licenses.

E.4 Technology Levels and Ports

MSMC Fault Management licensing is based on Technology Levels and Ports. Furthermore OpenScape FM provides the collected number of Ports according to the Technology Type.

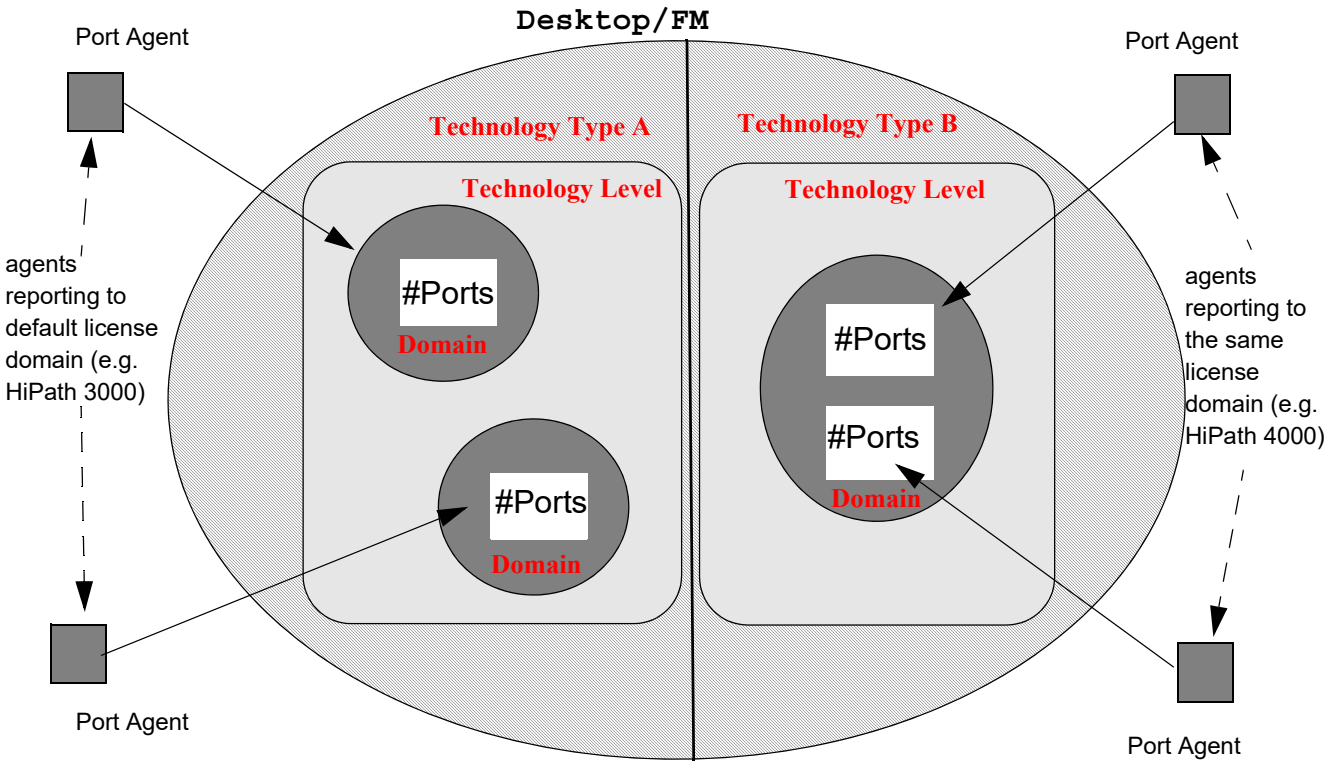
The notions Technology Type and Technology Level are used with the following meaning:

- **Technology Type**
  Is used to distinguish between nodes/devices of different system families which are managed by the Desktop and Fault Management. Examples for system families are HiPath 3000/5000 and HiPath 4000.
- **Technology Level**
  Is used to aggregate a defined range of Versions within a specific Technology Type into one Level / set. Hence, distinct Versions can be assigned to one Technology Level. New versions of a Technology Type (e.g. HiPath 4000 systems) have a higher Technology Level than older versions (e.g. Hicom 300) of the same Technology Type.
- **Port**
  A physical Port belongs to a certain Technology Type and Technology Level of that type. It is the basic unit in the license verification process.
- **Domain**
  Is used to avoid a double count of Ports in Hot Standby solutions and in assistant/manager scenarios. If two HiPath Port Agents report Ports belonging to the same license domain, they will not be counted twice (see Section E.5, “Port Manager”). If the agents report different amounts of Ports, the Port Manager will use the maximum number of Ports reported for a license domain.

The OpenScape DT contains a module “Port Manager” which is responsible for the collection and processing of Port License data (Appendix E.5, “Port Manager”). The Port License data is retrieved from the HiPath Port Agents. A HiPath Port Agent has to be supported by each HiPath technology taking part in the Port based licensing process. As a consequence, in a HiPath environment where several different HiPath Technologies exist the corresponding Port Agents have to be active to enable license verification.
In the OpenScape DT GUI a HiPath Port Agent is represented by a Port Collector object. The Port Collector symbol is located below the IP Node for which the corresponding Port Agent was discovered. A Port Collector object offers the menu item **Domain Ports** opening a browser which provides information about Domains, Ports, Technology Levels and Technology Types.

Each plugin operating in FM mode will check the actual number of managed Ports once a day. If the number of Ports licensed is exceeded a license violation occurs. Such a violation will be handled in according with the rules described in Appendix E.6, “License Manager”.

The conceptual view of the OpenScape Desktop/FM concerning Technology Types, Levels, Ports and Domains is illustrated by the following figure.



E.5 Port Manager

The HiPath Port Manager acts as the central information and collecting instance for all Port license data. It discovers the HiPath Port Agents and creates for each discovered agent a Port Collector object. The Port Collector objects retrieve the Port entries (Technology Type, Level, Port number, Domain id) from the Port Agents and provide this information for further processing to the Port Manager. All IP nodes for which a Port Agent has been discovered are presented on the submap below the Port Manager.

Further, the Port Manager supports license Domains. A license Domain is used with Hot Standby HW configurations and with assistant/manager configurations to avoid double counting of licensed Ports. A Domain is identified by a License Domain id. If two discovered Port entries match in the values domain id, Technology Type and Technology Level, the Port number is counted only once for the related license domain. If two Port Agents provide different Port numbers for the same Domain, Type and Level, the Port Manager uses the information of the Port Agent reporting the higher port number.

In the case a Port Agent does not provide a Domain id with the Port entry, the Port Manager will use the “(ip-address)” of that Port Agent as a default domain id.

A license domain will be set up by the HiPath device/system supplier.

If the Port Manager recognizes changes of the Port information it re-reads all Port information.

The symbol of the Port Manager is to be found in the OpenScape DT GUI below

Root->System->Server

The Port Manager symbol provides the following menu items:

Overall Port Sum... : Provides the sum of Ports of all registered Ports, independent from a Technology Type or Level. The license domains, however, are included in this calculation.

Ports By Technology Type... : Provides the sum of Ports of all Port Collectors sorted by Technology Types.

Ports By Level... : Provides the sum of Ports of all Port Collectors sorted by Technology Types and their Levels.

Ports By Domain... : Provides the sum of Ports of all Port Collectors sorted by Technology Type, Levels and domains. The default domains are set to “(IP-Address)“ to provide a better overview.

E.6 License Manager

Technology Type and Level specific license checking will be performed automatically:

- once a day,
- each time a license file is entered and
- each time the server is started.

Furthermore a user can manually perform a license check by requesting the license information from the License Manager (see Section E.6.1).

First of all it will be checked whether a license for the requested feature is available. Available means:

- the corresponding license could be checked out successfully from the CLA,
- the corresponding license feature is covered by the license file (only for HPQM, HPUM),
- the time period defined by the license feature is valid,
- the license information is not corrupted
  (e.g. the attribute string format has to be valid, the digest has to match the license string)

The licenses will be checked against the IP address of the OpenScape FM server or against the MAC address of the Network Interface Card used by the OpenScape FM server, depending on the licensing method used (only for HPQM, HPUM).

Then the validity of the Fault Management licenses for all plugins will be checked guided by the following basic rules:

- Free Port Licenses of a higher Technology Level can be used to satisfy missing licenses for Ports belonging to a lower Technology Level independent of Technology Type. However, it is not possible to use free Port Licenses of a lower Technology Level to satisfy missing Port Licenses in a higher Technology Level.

Example:
Hicom 300 and HiPath 4000 systems belong to the same Technology Type. The Ports of Hicom 300 systems are assigned to Level “1” and the Ports of HiPath 4000 systems are assigned to Level “2” and both are assigned to the same Technology Type “HiPath 4000”. Let’s assume that, the MSMC FM has 1000 Port Licenses for HiPath 4000 Level 1 and 1000 for HiPath 4000 Level 2.

The following Port License allocation is valid:

Level 1 - Hicom 300: 1500 ports (note: 500 license ports are allocated from Level 2 ports)

Level 2 - HiPath 4000: 500 ports

The following Port License transfer will not be possible:

Level 1 - Hicom 300: 500 ports (note: 500 ports of this level are not used)

Level 2 - HiPath 4000: 1500 ports (note: 500 ports of level >=2 are needed)

A license violation is detected when the number of managed Ports exceeds the number licensed Ports. The reaction of license violation and failure is described in the following sections:

- Section E.6.2, “Reaction on Absent Licenses”
- Section E.6.4, “Reaction on Technology Type specific license violations”

E.6.1 Checking the License Status

A user can check the license status via the OpenScape DT Client using the main menu item **Server->Administration->License Manager->License Information** and the page **License Status**. Opening the dialogue will perform a license check, Section E.6. The following information will be displayed in the opened dialogue:

- **Feature:** name of the feature.
- **State:** state of the license feature, see Table 1.
- **Last Check:** date of the last license check.
- **Start Date:** starting date of the license period.
- **End Date:** expiring date of license period
- **Licensed Version:** the version of the plugin/application the license is granted for.
- **Requested Version:** the version of the installed plugin/application the license is requested for.
- **Days until Expiry:** days remaining until the license will expire.
- **Grace Period:** if a violation warning has occurred it shows the days remaining until a violation error will be reported.

- **Attributes:** contains feature specific information. E.g. number of Ports licensed for a specific Technology Level. Different Technology Levels are separated by a “|“ and the dedicated Ports by a “;“. For example the Attributes “1;500|2;1200“ means that for the Technology Level “1“ 500 Ports are licensed and for the Technology Level “2“ 1200 Ports.
- **Description:** description of the license.

For each license feature checked, an object will be created under the License Manager. The License Manager is located at **Root->System->Server->Administration->License Manager** in the object hierarchy. The feature object offers the menu item **Information** in its context menu. This menu item shows the same information as listed in the “License Status“ information browser but only for this feature. Each feature symbol has a state. The severity (indicated by a color) of the object state depends on the result of the last license check (the different severities are described in *Section 5.11, “Topologies in the Submap and Info View Area”*):

License Status	Severity	Description
Ok	NORMAL -> green	The license is ok.
Feature Not Found	UNKNOWN -> blue	The license feature requested was not found in the license file.
Expiration Warning 1	WARNING -> light blue	The license ends in 60 to 31.
Expiration Warning 2	MINOR -> yellow	The license ends in 30 to 21.
Expiration Warning 3	MAJOR -> orange	The license ends in 20 to 11.
Expiration Warning 4	CRITICAL -> red	The license ends in 10 to 1.
License Expired	DISABLED -> dark brown	The license period of the feature has expired.
License Not Started	DISABLED -> dark brown	The start date of the license has not yet been reached.
Digest Error	CRITICAL -> red	The digest does not match the license information.
Host Error	CRITICAL -> red	The feature is not licensed for this host. (IP address based licensing only)
MAC Error	CRITICAL -> red	The feature is not licensed for this host. (MAC address based licensing only)
Info Error	CRITICAL -> red	The info string does not have the expected format.
Version Error	CRITICAL -> red	The requested license version does not match the licensed version. This is the case when the major version requested in a license is greater than the major version of the license information.
Invalid Format Error	CRITICAL -> red	The license feature has an invalid format.
Violation Warning	MINOR -> yellow	A violation for the license feature has been detected. The license feature is now in the state “violation warning” for a grace period of 30 days, see Appendix E.6.4, “Reaction on Technology Type specific license violations”.

Table 1 License Status Severity Mapping

License Status	Severity	Description
Violation Error	MAJOR -> orange	A previous license violation grace period has expired and the license feature is now in the state “violation error”, see Appendix E.6.4, “Reaction on Technology Type specific license violations”.

Table 1 License Status Severity Mapping

To get detailed information about the current usage of the licenses, the menu item **Server->Administration->License Manager->License Information** opens a browser offering the following information on the page **Detail Information**:

- **Feature:** Name of the feature.
- **Technology Level:** Level of the licensed Technology.
- **Number:** Number of collected Units.
- **Max. Number:** Number of maximum licensed Units.
- **Unit:** Unit, which is licensed, e.g. Ports, IP Nodes or EPM MIB objects.
- **Check Ok:** Indicates if the License Check results in an ok or not.
- **Ports of Technology:** Shows the collected number of units according to the Technology Type.

Opening this dialogue will also perform a license check.

E.6.2 Reaction on Absent Licenses

When the OpenScape DT is started and the corresponding license feature is not accessible or not valid, the following limitations will occur:

- **OpenScape DT:** All Fault Management functions are deactivated.
- **IP Manager:** It is not possible to add additional IP Nodes.
- **Enterprise MIB:** It is not possible to add additional EPM objects, EPM main menu, MIB Browser specific menus/functions and Threshold specific menus/functions are deactivated.
- **MAR:** The functionalities of the MAR plugin are deactivated, but former defined reactions will still be fired.
- **HiPath 3000/5000 FM**, **HiPath 4000 FM**, **HiPath MIB FM**, **IP Manager FM, Enterprise MIB FM, OpenScape Voice:** The event browser will not display fault events, the graphical fault status (icon status color) of the devices/nodes/objects will be disabled and the Fault Management specific menus of the corresponding plugin are deactivated and Fault Management specific objects of the corresponding plugin are deleted.
- **Other Plugins:** If the license is missing or out of date, the plugin cannot be initialized or the functions are no longer available. If parameter limits are exceeded, the plugin can still be used for a grace period. If the parameters are exceeded after this period, the plugin will be deactivated.

In the case that a license was checked out from a license server and this server cannot be reached, the OpenScape Desktop will continue to run for a period of 30 days. After that period, it will stop if the license server is not restarted. If the license server starts again and the OpenScape Desktop license (in the course of a license check) could not be checked out, the OpenScape Desktop will stop.

E.6.3 Reaction on IP Address or MAC Address Change

Depending on the license type used an IP address or MAC address change has different effects:

- **CLA:** If the IP address or MAC address found in the license feature does not match the IP address or MAC address gathered by the system, a 30 day grace period is started. If this grace period has been expired, the corresponding feature will be deactivated.
- **Internal:** If the IP address or MAC address found in the license feature does not match the IP address or MAC address gathered by the system, the corresponding feature will be deactivated immediately (only for HPQM, HPUM).

E.6.4 Reaction on Technology Type specific license violations

HiPath 3000/5000 FM, **HiPath 4000 FM**, **HiPath MIB FM**, **IP Manager FM, OpenScape Voice:** If the number of active Ports discovered by the OpenScape DT exceeds the number of licensed Ports for a specific Technology Level (a so called “violation warning“), a 30 days grace period is started. During the grace period the functionality of the MSMC system will not be restricted. A warning “Number of active Ports exceeds number of Ports licensed” event is generated every day during this grace period.
At expiration of this grace period (“violation error“) all fault specific menus of the HiPath 3000/5000 FM, HiPath 4000 FM, HiPath MIB FM, and the IPManager FM plugin are deactivated. However, the fault specific events and the fault status of the corresponding objects will still be displayed. This mode is called the restricted FM mode. Thus the Fault Management will be activated by one license feature, a license violation has impact on all plugins which were in Fault Management mode.

Important Note:
If no Port Agent is found for a HiPath 4000 system, the fault specific menu entries will be deactivated for the corresponding objects.
If a Port Agent for a HiPath 4000 system is not accessible for more than seven days, the fault specific menu entries will be deactivated for the corresponding objects, too.

E.7 License feature information in the Logo Area

Some license features offer information which will be displayed in the Logo Area of the client, when it is connected to the server. So each user can see which functions are licensed and active. When the license feature is expired the information will no longer be displayed.

F Obtaining a License Key

To run the OpenScape FM you need a valid license. If you are trying to start the OpenScape FM without a valid license, the OpenScape FM will not start.

To obtain a valid license key please contact Unify.

OpenScape FM Desktop and the IP Manager are shipped with a demo license each, please refer to *Appendix E, "Licensing"* to get more details about the licensing of OpenScape FM.

G Server Update

If OpenScape FM is already installed and an update to a new version should be performed, this can be done by simply installing the new version (including the bundled Performance Management) into the existing OpenScape FM directory.

Manual or even automatic updates can also be performed by using a Unify SWS Server (see *Section G.1*).

Important Note:
An upgrade to OpenScape FM V11 is only supported from OpenScape FM V10. Upgrades from older versions have to be performed incrementally.

A valid license for the new product version has to be provided/installed via the CLA.

Platform Change:

If the OpenScape FM server should be moved to another platform, e.g. from Windows 2k to a Unix machine, or to another system with a new IP address, the relocation can be performed without data loss. The OpenScape FM can be installed on the new platform and the old OpenScape FM database can be inserted. It is suggested to set up the new server first and to test it before the old server is removed.

Important Note:
The new system should be unknown by the OpenScape FM.
In other words: It should not have already been monitored by the old server.

The following steps should be performed on the new Server:

- Installation of OpenScape FM on the new platform as explained in *Appendix C, "Installation Process"*.
- Login to the OpenScape FM Server via the client.
- Initialization of the required plugin modules.
- Installation of the license file or a connection to the CLA for the OpenScape FM and the required modules.
- Stopping of the OpenScape FM Server.
- Replacing the database files in the new OpenScape FM installation:
  Removing the database files which have been installed and copying the old database files into the database directory.
  Windows: `<Default Install Dir>\server\database`
  Unix: `<Default Install Dir>/server/database`
- Starting the new Server.
- Starting the OpenScape FM standalone Client or login to the OpenScape FM Server via web browser.

G.1 Updates via Unify SWS Server

To keep the OpenScape FM software on a current level, a software update function is provided. If enabled, this update function uses the Unify SWS to check on a regular basis whether new OpenScape FM or System Management Agent (see *System Management User Guide*) loads or patches are available. The check can also be triggered manually.

According to the configuration of the update function, new loads or patches can be loaded and installed automatically.

Important Note:
To use this update function, the OpenScape FM server must be able to connect to the internet and the Unify SWS server. To perform an update, the OpenScape FM has to be registered to the Unify SWS.

G.1.1 Update Configuration

The configuration page of the software update function can be opened within the OpenScape FM Administrator Console by using the main menu entry **Server->Administration->Server Properties**. It is located on the page **Update** which displays the following lines:

- **Update Server**
  The address of the Unify SWS update server is known by the OpenScape FM. If and when a connection to the update server was successful is displayed in this line. The button **Configure proxy-server** can be used to configure of a Proxy Server to reach the Unify SWS.
- **Registered**
  This line shows if and when a registration to the update server was successful. The button **Register** can be used to define the authentication parameters (**Login** and **Password**) that will be used for the registration.
- **Auto Check**
  If this line is checked, the update function performs automated checks for new loads and patches. The line also shows the available load or patch version. The button **Check** can be used to manually trigger an immediate check for updates.
- **Check every**
  This line is used to configure the time interval in which automatic checks for new loads and patches should be performed. For example, checks can be performed every day at 3pm.
- **Auto Download**
  If this line is checked and a new load or patch is detected, a download for the load or patch will be automatically initiated. The line displays the version of the last downloaded load or patch, and the button **Download** can be used to initiate a download manually.
- **Auto Install**
  If this line is checked and a new load or patch is automatically loaded, this load or patch will be automatically installed. The line displays the last load or patch version that has been installed. The button **Install** can be used to manually start the installation of the last downloaded load or patch.

- **Backup**
  If this line is checked, the current content of the OpenScape FM installation directory (with the exception of the Java DB) is saved before an update is performed. The storage location and the name of the storage archive can be selected using the input field and the **Browse** button.
- **Free Disk Space Requirement for Update**
  In this line it can be specified how much disk space must at least be available on the installation partition for an update to even begin. A sufficiently large volume should be specified here, especially in connection with a backup.
  If a backup is aborted due to lack of space, a corresponding event is displayed in the OpenScape FM.

If an OpenScape FM load or patch is installed that contains loads or patches for its **System Management Agents** these agents can also be updated automatically. The *Internal System Management Agent* will always be updated as part of the OpenScape FM update, but the *External Agents* that should be updated have to be configured individually.

The configuration page for External System Management Agents can be opened by using the main menu entry **System Management->Update**.

The page lists all External System Management Agents in separate table rows which contain the **Server Name**, **Server IP** and currently installed agent **Version**.

The check boxes in the column **Auto Update** define which of the agents should be automatically updated, if a new agent load or patch has been found during the installation of an OpenScape FM load or patch.

The button **Update** on the bottom of the table can be used to manually install the latest load or patch to the selected agents.

G.1.2 Update Execution

The OpenScape FM uses a separate Windows or Linux service **OpenScape Update Service** to install new loads or patches.

This service gets registered during the installation of the OpenScape FM and it will be started by the OpenScape FM when a new load or patch has to be installed.

The Update Service invokes a predefined command with options to perform a silent load or patch installation. When the installation is finished, the Update Service terminates. The OpenScape FM Startup Service gets stopped and started by the load or patch installation.

The System Management Agents also include an update service.

If **Auto Update** is enabled for an External System Management Agent and a new load or patch has been downloaded, the load or patch is transferred to the agent and it gets instructed to perform the installation of this load or patch. The agent‘s update is performed by starting the update service of the affected agent.

If **Auto Update** is not enabled for an External System Management Agent, updates for this agent have to be performed manually.

The transfer of the new load or patch to an agent is performed by using the already known/used ports of the System Management Agent. It is therefore *not* necessary to add additional ports to existing firewall rules.

Server Update

Updates via Unify SWS Server

Index

N

O

P

Q

R

U

V

W