HP OpenView Service Assurance for Communication Networks

Customization and Maintenance Guide

HP-UX, Solaris, Windows NT®

Manufacturing Part Number: J5119-90005

October 2001

Legal Notices

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

HEWLETT-PACKARD COMPANY
United States of America

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

Acrobat® is a trademark of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Contents

1. OV Topology Server Management Overview

Administrative Tasks 23
Topology Server Management 23
Database Management 23
Topology Object Management 24
User Environment Customization Management 25
Planning the Site Administration. 26

2. Managing Topology Servers

Managing the FM Servers. 31
Starting the FM Servers 31
Verifying the Status of FM Server. 33
Listing All FM Servers in The Installation. 34
Recovering the Problem Database. 35
Shutting Down the FM Servers. 36
Managing the GUI Servers 38
Starting Up GUI Servers 38
Verifying the Status of GUI Server 39
Shutting Down GUI Servers 39
Managing OV Topology Server Services. 41
Stopping OV Topology Server Services 41
Rebooting 41
Starting OV Topology Server Services 41
Handling All Installed Servers 43

3. Managing the OV Topology Server Databases

OV Topology Server Databases 47
CORBA 49
Managing CORBA Hosts 49
Managing Users 50
Overview of Unified Services 51
Managing CORBA Databases 54

Contents

CORBA Data Directories 54
CORBA Database Concepts and Administration 55
Backup and Recovery 57
Time Synchronization 63
Managing CORBA Users 64
Listing CORBA Users 65
Adding, Removing, and Listing Host-Specific Logins 65
Backup/Recovery of FM Server Databases 67
Physical Backup/Recovery 68
Database Housekeeping at Application Level 70
Maintaining the Presentation Database 79
Backing up the GUIDB 79
Restoring the GUIDB 79
Physical Backup of GUIDB 80
OV Topology Server Backup Schedule 82
Changing Oracle User Passwords within OV Topology Server 85
Using oemfpasswd 86
. 86

4. Managing a Project

Introduction 89
Validating a Project 90
Deploying a Project 91
Applying a Project 92
Removing Topology Data 94
Redeploying a Project 95
Restoring Backup Configuration Files 96
Removing Backup Configuration Files 97

5. Managing Topology Objects

Managing Object Classes 101
Adding Managed Object Classes 101

Contents

Removing Managed Object Classes 102
Updating Configuration with New Topology Model 103
Managing Topology Instances 104
Adding Managed Object Instances 104
Removing Managed Object Instances 105
Updating Topology Instance Data 106
Managing Agent Configuration 107
Adding Data Collectors 107
Removing Data Collector Definitions 112
Adding Lower Topology Instances 115
Removing Lower Topology Instances 118
Updating Agent Configuration 118
Managing OM Events 120
OM Event Generator 120
Format of the Intermediate File 120
Generating OM Events 123
Managing Topology Maps 125
Using the Map Presenter to Manage Domains 125
Creating Link Object Submaps 126

6. Configuring Operator Environments

About the iNOC Console 133
Prerequisites 133
Smart Navigations 133
Users and Operation Profiles 134
Topology GUI Login 135
About Configuring Operator Access 137
Operation Profile Configurator 137
Default Operator and Operation Profile 139
Configuration Files for Users and Profiles 140
Adding iNOC Users 142
Adding OVO Users 142

Contents

Adding a Topology GUI User 143
Adding an OS User 143
Deleting a User 144
Configuring Operator Access 146
Define the Managed Object Domains. 148
Define Management Domain Groups (optional) 151
Define Problem Filters 157
Define Operation Profiles 165
Assigning Profiles to Users 177
Associating Operation Profiles with Users 177
Customizing Topology GUI Login 178
Adding Authentication 178
Changing Topology GUI profiles 179

7. Customizing the Topology GUI

Common Administrative Tasks 183
Using the Admin Panel 184
Starting the Admin Panel. 184
Read-Only Information in the Admin Panel 186
Change Ownership to Maps and Backgrounds. 191
Changing Ownership of a Map. 191
Copying a Map for a New User 192
Adding Map Backgrounds 193
Additional Map Information 193
Define Map Symbols. 196
Predefined Map Object Classes 196
User-Defined Map Object Classes 200
Create Mappings for Colors or Bitmaps 206
System Mapping 207
Color Mapping. 208
Mapping Bitmaps 208
Changing the Perceived Severity Color Mapping 209

Contents

Adding a New Color Map 210
Using Bitmap Mappings in the Topology GUI 211
Configuring Local Form Probable Cause Mapping. 212
Configure Blinking Objects 215
Configure Blink Expiry Time. 216
Set Colors, Fonts, and Alarm Bell for the GUI Client 217
Defining Colors 219
Defining Fonts 220
Defining Fonts 221
Setting Alarm Bell 221
Define Menubars 222
Defining the Parameters 224
Defining the Menu Positions 225
Adding a New Menu Command. 227
Configuring Save Session for Other Roles 227
Define Toolbar Buttons 229
Defining the Parameters 231
Defining the Toolbar Button Positions 231
Set New Actions for Actions Menu 233

8. Other Administrative Utilities

Administering Event Correlation 239
Establishing Logical Connectivity. 239
Updating Correlation Rules. 240

9. Troubleshooting

Verifying Server Status 243
Verifying the Status of the Topology Server 243
Understanding Errors Listed in the syslog 244
FM Server Does Not Startup After System Re-boot 245
Issues Related to Problem Presenter Display 246
Shortname *unknown* in Problem Presenter Display 246

Contents

Incorrect Information in Problem/Map Presenter. 246
Alarm Sent to Non-Existing Object 247
Unlocking Sessions. 248
Monitoring and Resolving Replication Conflict 249
Avoiding and Resolving Replication Conflicts in Map 249
Using the Admin Panel to Avoid and Resolve Replication Conflict . 250
Avoiding and Resolving Replication Conflicts in User Session 250
Troubleshooting CORBA Problems 251
CORBA Problems / Solutions 252
Searching and Managing Logs 253
Searching HP OVC CORBA Log Messages 253
Searching Other Error Logs 255
Managing HP OVC CORBA Message Logs 255
Troubleshooting Topology GUI Related Problems 257
Problem Presenter Not Being Updated 257
Setting the Background Map for the Map Presenter 257
Alt Key Does Not Work. 258
Using Trace Logs 259
Using Trace Logs on the FM Server 259
Using Trace Logs on the GUI Server. 262
Using Trace Logs on the Topology GUI 262
Troubleshoot Topology Database Inconsistencies 265
Syntax 265
Command Line Programs 269
OV Topology Server Tables 279
Table - alarm 279
Table - alarm_additional_info. 279
Table - problem_to_event_map. 280
Table - problem_state 280
Table - ec_state 280
Table - problem_audittrail 280
Table - alarm_additional_text 280

Contents

Table - problem_annotation 280
Table - alarm_state_change 281
Table - alarm 283
Indices 285
Table - alarm_additional_info 286
Indices 286
Table - problem_to_event_map 287
Indices 287
Table - problem_state 288
Indices 289
Table - ec_state 290
Indices 290
Table - problem_audittrail 291
Indices 291
Table - alarm_state_change 292
Indices 292
Table - alarm_additional_text 293
Indices 293
Table - problem_annotation 294
Indices 294
List of Applications and Tasks 297

Contents

Tables

Table 3-2. Database Management Command-Line Tools 54
Table •. Storage Methods Used by CORBA Services. 56
Table •. File 58
Table •. Tablespace 59
Table •. Tablespace 59
Table NOTE. User Management Service Command-Line Tools. 64
Table •. Recovery Implications 70
Table NOTE. Database Backup Suggestions 82
Table NOTE. Description of Apply Commands 92
Table 5. Adding MOCs Table. 101
Table NOTE. Removing MOCs Table 103
Table •. Adding Upper Topology Instances 104
Table •. Adding Connection Instances 105
Table 5-4. Removing Upper Topology Instances 106
Table •. Adding DCs 108
Table •. Adding Sources 109
Table •. Adding Source Details. 110
Table 5-8. Adding Equipment List 110
Table 5-9. Adding Record Formats 111
Table 5-10. Adding Lookup Tables 112
Table •. Removing DCs 112
Table IMPORTANT. Removing Sources 113
Table 5-13. Removing Source Details 113
Table 5-14. Removing Equipment List 114
Table 5-15. Removing Record Formats 114
Table 5-16. Removing Lookup Tables 115
Table •. Adding Network Element Instances 115
Table •. Adding Managed Object Instances. 116
Table •. Adding Connection Instances 117
Table NOTE. Removing Managed Object Instances 118
Table NOTE. Import File Field Description. 121
Table 6-3. Managed Object Domains 148

Tables

Table 2. Management Domain Group 151
Table •. Problem Filters. 157
Table 2. Operation Profiles 165
Table •. Work Schedule 175
Table 7-4. Attributes of Presenter Type 189
Table 7-6. Column 194
Table 7-8. DisplayType Attributes 199
Table NOTE. Attributes to Describe Map Object Classes 200
Table 4. Attributes for the Map Object Class 202
Table 7-9. Classes Datatype Fields 203
Table 7-11. Column 207
Table 2. Sample Labels for Default PerceivedSeverityColorMap 209
Table 11. Admin State Numerical Values. 211
Table 6. Local Form Probable Cause Example 213
Table 7. Local Probable Cause Codes Not Needed 214
Table 8. Values for the Mapping Element OVCSABlinkConfigMapping . 216
Table 7-12. Column 218
Table 7-15. Column 223
Table 5. Save Session MenuBar Settings 228
Table 7-18. Column 230
Table 7-21. Column 234
Table •. Message Log Searching Command-Line Tools. 253
Table •. Message Log Management Command-Line Tools 255
Table B-1. Column Name. 283
Table B-1. Column Name. 286
Table B-1. Column Name. 287
Table B-1. Column Name. 288
Table B-1. Column Name. 290
Table B-1. Column Name. 291
Table B-1. Column Name. 292
Table B-1. Column Name. 293

Tables

Table B-1. Column Name 294
Table •. Tasks Associated with managed_object_application 298
Table •. Tasks Associated with problem_application 298
Table 2. Tasks Associated with outage_plan_application 301
Table 2. Tasks Associated with om_event_application 302

Tables

Figures

Figure 3-1. Two Sample Models of Installations 50
Figure 3-2. The Unified Service Model. 53
Figure 6-1. Relationship Between GUIs. 134
Figure 6-2. Operation Profile Main Menu Screen 138
Figure 6-3. Steps in Planning Operator Access to Topology GUI 146
Figure 6-4. Managed Object Management Domain Maintenance Window 149
Figure 6-5. Management Domain Group Maintenance Window 152
Figure 6-6. Management Domain Group Association Window. 153
Figure 6-7. Problem Filter Maintenance Window 159
Figure 6-8. Problem Filter Attributes Window 161
Figure 6-9. Operation Profile Maintenance Window. 166
Figure 6-10. Operation Profile Modification Window 167
Figure 6-11. Application Association Window 170
Figure 7-1. Admin Panel 184
Figure 7-2. Users Data Panel 187
Figure 7-3. Roles Data Panel 188
Figure 7-4. Presenter Type Panel. 189
Figure 7-5. Map Datatype Showing New Map Ownership 192
Figure 7-6. Map Datatype Panel. 194
Figure 7-7. Graphic Images for Predefined Map Object Classes 197
Figure 7-8. Predefined Map Object Classes 198
Figure 7-9. Complete Class Specification. 203
Figure 7-10. Function to Icon Mapping in Mapping Datatype 204
Figure 7-11. Mappings Panel 206
Figure 7-12. Resources Panel 217
Figure 7-13. Color Chooser Panel. 219
Figure 7-14. Font Chooser Panel. 220
Figure 7-15. Menubars Panel 222
Figure 7-16. Parameters Dialog 225
Figure 7-17. Position Dialog Box 226
Figure 7-18. Toolbars Panel 229

Figures

Figure 7-19. Parameters Dialog. 231
Figure 7-20. Position Dialog Box 232
Figure 7-21. Actions Datatype 233
Figure 7-22. Available Classes. 235
Figure B-1. Link Between the Tables 282

In This Book

This book describes the administrative functions of HP OpenView Service Assurance for Communication Networks. It describes how to:

- Start and shut down the OV Topology Server servers.
- Check the status of the servers.
- Troubleshoot the installation for its administrative functions.
- Load and maintain object information details.
- Load and maintain GUI presentation details.

Audience

This manual is intended for the network administrator. It assumes that the administrator is familiar with HP-UX operating system and Oracle RDBMS. It also assumes that the administrator is familiar with using GUI-based applications with mouse and menu-driven interfaces on UNIX workstations and the Windows NT operating system.

Manual Organization

This book contains the following chapters:

Chapter 1: Overviews the administrative tasks for OV Topology Server.

Chapter 2: Describes the procedure to start up, run, and shut down the topology server. It also describes the CORBA administrative utilities in the context of OV Topology Server.

Chapter 3: Describes the database management procedures, including the processes for backing up and restoring the databases maintained on the topology server machines and the procedure to change Oracle passwords within OVSACN.

Chapter 4: Explains the MOI Handler utility that loads and updates managed object instance information. The MOI Handler imports and exports the managed object instances in the form of flat files to and from the Map Presenter. These procedures can be used to load the Map Presenter and back up the map object instances information, respectively. This chapter also explains how to add a new object and link it up to the existing network.

Chapter 5: Describes the utility to generate OM Events.

Chapter 6: Describes the administrative tasks enabled for the topology GUI interface and the procedure for customizing the operator interface.

Chapter 7: Describes utilities for managing event correlation.

Chapter 8: Describes utilities for troubleshooting problems that may be encountered during the use of the product. This chapter also describes the utilities that check installation information, such as the version details of the servers in the installations.

Appendix A: Details the error messages that may be encountered while using the product, including the cause and corrective action to take.

Appendix B: Provides details of the database schema (in which the problems are logged) used in the product.

Appendix C: Provides the list of applications and tasks for which access rights can be assigned to the product operators.

1 OV Topology Server Management Overview

OV Topology Server Management Overview

This chapter introduces the OV Topology Server and defines the concepts and terminology used.

It overviews the administrative tasks to help network administrators ensure that the product runs effectively and efficiently.

All tasks described in this manual should be executed by the network administrator unless otherwise mentioned. The default login ID of the administrator is `oemfadm`.

Administrative Tasks

This section describes the administrative tasks that can be performed on the OV Topology Server. These tasks can be categorized as:

1. Topology server management
2. Database management
3. Topology object management
4. User environment customization management

This chapter explains these categories of administrative tasks. The following chapters describe the utilities used to execute these tasks.

Topology Server Management

Topology server management covers administrative tasks that pertain to maintaining the components of the OV Topology Server. Each of the following tasks is described for each server individually, as applicable:

- Starting up the servers.

  Each server must be individually started up.

- Shutting down the servers

  The server shutdown process is configured as part of system shutdown. Therefore, the server need not be separately shut down before system shutdown.

- Checking the status of the servers.

  The system administrator can check the status of individual servers to ensure that all server processes are running.

Database Management

- Configuring OV Topology Server databases.

  The alarm and object database tables can be configured independently of the server installation process. This independence enables the definition and modification of the OV Topology Server databases as required for individual Fault Management (FM)

Servers. The system configuration utilities, which enable this function, must be run as the first step to setting up the topology server after the FM Server product has been installed on the host.

- Managing OV Topology Server databases.

  The following utilities manage the OV Topology Server databases:

  — Housekeeping

    A set of utilities enables selective backup and restoration of records in these databases:

    — Problem Database: `fmsalmbackup` and `fmsalmrestore`

    — OM Event Database: `fmsomeventbackup` and `fmsomeventrestore`

    — Outage Plans Database: `fmsoutagebackup` and `fmsoutagerestore`

    — Presentation Database: `guidbbkup` and `guidbrestore`

    The records are backed up and, optionally, deleted from the database when running each backup utility.

    These utilities operate on the local database only, that is, on the database of the location where the utility is executed. They backup and restore only the logical databases. Details on how to run these utilities are provided in “Backup/Recovery of FM Server Databases” on page 67, “Restoring Application Level Backup” on page 75, and “Maintaining the Presentation Database” on page 79.

  — Problem Database Recovery.

    The `fmsalmrecover` utility enables recovery of the Problem Database after a system crash or any abnormal system shutdown. This recovery utility can be executed while the FM Server is running.

    For details on using this utility, see “Recovering the Problem Database” on page 35.

Topology Object Management

- Adding/removing topology objects

  The number of managed object classes that can be created depends on

the installation model; this number can easily run to the thousands.

- Adding/removing topology object instances

  The number of managed object instances that can be stored in the topology database depends on the installation model; this number can easily run to the hundreds of thousands.

- Topology object import and export

  The `fmsmhimport` utility inserts managed object instances into and removes managed object instances from the topology database.

  The `fmsmhexport` utility exports information in the topology and presentation databases to an ASCII file.

  For details on using these utilities, see Chapter 5 , “Managing Topology Objects,” on page 99.

- Generating OM events

  The OM event generator is a command line utility that generates object management events from an ASCII file. This utility enables OM Event information to be imported to an event presenter window.

User Environment Customization Management

The customization of user access and display properties is facilitated through the Admin Panel on the topology sever machine.

A network administrator can configure the operator environment by performing the following tasks:

- Assigning operator profiles to users.
- Create or modify symbols used to represent map objects.
- Create color mappings or bitmap mappings for problem, outage, and OM event tables.
- Define the mapping of blinking objects on the map presenter.
- Modify the operator's view by specifying presenter colors or fonts or setting the alarm bell.
- Define the visibility and location of menu commands.
- Define new actions and/or create menu commands to access those actions.

Planning the Site Administration

To ensure smooth administration of the site, the following administrative commands and utilities are provided:

- Basic administration commands for server startup, shutdown, and status checks.

  These commands enable the start up and shut down of servers individually and collectively. For the most effective use of OV Topology Server, the servers must be started up or shut down in a specific order.

  1. For the FM Server, including CORBA, see “Managing the FM Servers” on page 31.
  2. For the GUI Server, see “Managing the GUI Servers” on page 38.

- Populating the topology database.

  This task involves entering managed object instance information into the topology and presentation databases and creating link submaps for the connection object instances.

  The task of maintaining the topology object instances is described in “Managing Topology Instances” on page 104.

- Maintaining OM Event details.

  The OM Events generation command is explained in “Managing OM Events” on page 120.

- Maintaining the databases.

  OV Topology Server uses seven databases housed within four database instances. The recommended procedure for backing up and restoring these databases is described in Chapter 3 , “Managing the OV Topology Server Databases,” on page 45.

- Customizing user environment in OV Topology Server.

  Use the Admin Panel to customize the user environment. The tasks involved are explained in Chapter 7, “Customizing the Topology GUI,” on page 181.

- Troubleshooting OV Topology Server.

Initial troubleshooting procedures are explained in Chapter 9 , “Troubleshooting,” on page 241.

2 Managing Topology Servers

This chapter explains the utilities that enable you to:

- Start up and shut down the FM servers.
- Check the status of the FM server processes.
- Start up and stop CORBA processes.
- Startup and shutdown the GUI servers.
- Check the status of the GUI server processes.

Managing the FM Servers

Starting the FM Servers

At any location, the Fault Management (FM) server must be started and running before the GUI server and topology GUIs can be started. To start the FM server, log in as `root` and execute:

```
fmsstart
```

The FM server starts up the following processes/applications if they are not already running—Oracle, ORBPlus, CORBA services—before starting up the FM server modules. As the FM server is started up, it displays informational messages for each of its processes.

After the FM server is started, it is ready to receive problem messages from the telecom agents connected to it. After the GUI Servers are started up, operators can start their topology GUI applications.

NOTE

The FM server must be running before starting the GUI servers because the FM server is central to processing information from the GUI Servers.

In a non-high availability environment, if the system is rebooted prior to starting the FM server, stop the OV Server before starting the FM server.

When the FM server is started after one or more of its modules has shut down abnormally, the `fmsstart` utility shuts down all other modules and then starts up the server in recovery mode.

If the configuration files have been changed since the last startup, they are updated from the files in the new configuration directory on the local server.

When the startup process completes, `fmsstart` displays a status message. `fmsstart` logs messages to the `/var/adm/syslog/syslog.log` file.

Verifying the Status of FM Server

The utility to check the status of the FM server can be run by any user on the FM server machine. It displays the status of all processes of the FM server from which it is executed. However, if this utility is executed on the primary FM server, the status of remote FM servers can also be verified. To verify the status of one or more FM servers, execute:

`fmsstatus`

The format of the status message on the FM server is:

`modulename` *`status`*

Where:

`modulename` indicates the server process.

`status` indicates the status of the process. It could be `running`, to indicate that the process is running, or `not running`, to indicate the process has stopped.

For example:

```
Location: LOCAL HOST
Command: Getting status of FMServer
Output:
Interface to MD:                    running
Alarm Logger:                       running
Alarm Correlator:                   running
Problem Manager:                    running
Network Status Manager:             running
Managed Object Manager:             running
Managed Object Class Manager:       running
OM Event Manager                    running
Outage Plan Manager:                not running
Alarm Manager:                      running
```

This example lists all FM server modules.

In the above example, the Outage Plan Manager is not running. Check the `/var/adm/syslog/syslog.log` file for errors from the Outage Plan Manager. You can get a description of the problem using the `oemferr` utility. Correct the problem and run the `fmsstart` utility, if required. The system starts in recovery mode.

Listing All FM Servers in The Installation

Use the `/opt/OVCORBA/bin/ovsiteinfo` command to list the FM Server machines that make up an installation. The returned list reflects the hosts in the installation in which you executed the command.

For example, if you are logged in to host `mars` and there are two locations (mars and jupiter) in the installation, when you execute:

`/opt/OVCORBA/bin/ovsiteinfo`

the returned list is:

```
mars
jupiter
```

The `ovsiteinfo` command has no options.

Recovering the Problem Database

The problem database could be corrupted or made inconsistent through abnormal shutdown of the FM server or incorrect restoration of the problem database. The `fmsalmrecover` utility enables recovery of the problem database.

This utility can be run as part of the startup command using a recovery option or used as a standalone command run on an ad hoc basis. It can be run while the FM server is running. However, because this utility reads and writes into the problem database, it is recommended that it be run when the problem traffic is low.

To recover the problem database:

`fmsalmrecover`

The command displays informational messages on the screen. It displays the number of records that have been recovered in the `alarm` and `ec_state` tables and the total number of records recovered.

This command logs trace messages to `$FMSVAR/share/log/fmsalmrecover.log`.

Shutting Down the FM Servers

Shutting down the FM server means that alarms received by the telecom OVO agent in that location will not be processed into problems.

To shut down the FM server, log on as `root` on the server host and execute:

`fmsstop`

The `fmsstop` utility shuts down the FM processes on each server it brings down. When the shutdown process completes, `fmsstop` displays a status message. `fmsstop` logs messages to `/var/adm/syslog/syslog.log`.

While the FM server is shut down, the telecom OVO agent can continue to receive messages from the telecom devices. However, no update can be made from the topology GUI, nor can the topology GUI receive any updates of alarms or status change.

The telecom OVO agent maintains a cyclic buffer of a fixed size. If the FM server is shut down for a long time, the old messages are overwritten by new messages received after the buffer is full. This overwriting may result in lost messages.

When the FM server is started up again, the telecom OVO agent transfers messages from its cyclic buffer to the FM server.

Shutting down the FM server does not stop CORBA services, ORBPlus, and Oracle. For details on shutting down and starting up CORBA services individually, see the next section.

NOTE Because the `fmsstop` utility is part of the system shutdown process, the FM servers need not be separately shut down before system shutdown.

Managing the GUI Servers

Starting Up GUI Servers

The GUI Server must be started only after its host Fault Management (FM) Server is started and running. To start the GUI Server, log in as `root` on the GUI Server host and execute:

```
guisstart
```

This command starts the GUI server processes and displays messages on the screen as each process starts.

After the GUI Server starts, operators with workstations connected to it are able to launch the topology GUI from OVO.

NOTE

If the FM server and the GUI Server are on the same machine, you can use the `oemfstart` utility to start the FM server and the GUI Server in sequence. For details, see “Handling All Installed Servers” on page 43.

If the two servers are on different machines, they must be started individually.

Setting the Number of Objects

The number of objects that can be viewed on a map on the map presenter is determined by the value of the `ILT_MAXOBJECTS_PER_MAP` environment variable on the GUI server through which the map presenter is invoked. By default, the maximum number of objects displayed on a map is 300.

To change the maximum number of objects viewed on a map:

1. Log on as `oemfadm` to the GUI Server machine on which you wish to change the limit.

2. Enter the lines:

   `export ILT_MAXOBJECTS_PER_MAP = `*`number`*

   Where *number* is the new limit on the number of objects to be displayed.

3. Start the GUI Server as described in “Starting Up GUI Servers” on page 38.

To permanently change the maximum number of objects viewed on a map:

Change the value of `ILT_MAXOBJECTS_PER_MAP` in the `/etc/opt/OEMF/V5.0/GUIS/oemf/util/guisenv` file.

Verifying the Status of GUI Server

To verify the status of the GUI Server processes, use the command:

`guisstatus`

The format of the status message on the GUI Server is:

```
User-handler daemon is operational.
Communication layer is operational.
```

Shutting Down GUI Servers

Before shutting down the GUI Servers, it is recommended that operators close their topology GUI sessions.

To shut down a GUI Server:

1. Log in as `root` on the GUI Server machine to be shut down.
2. Enter the command:

   `guisstop`

This command shuts down the GUI Server processes on the server and displays messages on the screen during the shutdown process.

Though the GUI Server is shut down, the FM server can continue to process message updates from the telecom agent connected to it.

NOTE Because this command is part of the system shutdown process, GUI Servers need not be separately shutdown before system shutdown.

Managing OV Topology Server Services

It is important to follow the correct sequence when stopping and restarting services in OV Topology Server.

Stopping OV Topology Server Services

Always execute shutdown of OV Topology Server components as `root` in the following sequence:

1. All GUI Servers. On each GUI Server machine, run the command:

   `/opt/OEMF/V5.0/GUIS/oemf/util/guisstop`

2. FM server:

   `fmsstop -l all`

3. CORBA Service (`ovcorba`):

   `/opt/OVCORBA/bin/ovcorba_admin -stop`

4. Notification Service (`ovnsls`):

   `/opt/OVNSLS/bin/ovnsls_admin -stop`

5. ORBPlus:

   `/opt/OVCORBA/bin/ovcorba_orb -stop`

6. Oracle (see Oracle documentation)

Rebooting

After successfully shutting down services, reboot systems normally.

Starting OV Topology Server Services

Always execute startup of OV Topology Server components as `root` in the following sequence:

1. Execute **`fmsstart`**, which automatically starts the following:

   a. Oracle (see Oracle documentation)

   b. ORBPlus (`/opt/OVCORBA/bin/ovcorba_orb -start`)

c. Notification Service (`/opt/OVNSLS/bin/ovnsls_admin -start`)

d. CORBA Services (`/opt/OVCORBA/bin/ovcorba_admin -start`)

e. FM servers (`fmsstart -l all`)

2. Execute **`/opt/OEMF/V5.0/GUIS/oemf/util/guisstart`** to start the related GUI Servers.

Handling All Installed Servers

Two utilities start up and shut down OV Topology Server and related products using a single command each. These utilities can only be run by `root` on the *server to start or stop*. The utilities are:

`oemfstart`	Starts up the topology servers and related servers or applications. This utility calls scripts that start up the appropriate utilities.
`oemfstop`	Stops the topology servers and related applications. This utility calls scripts that shut down the servers and related applications.

3 Managing the OV Topology Server Databases

This chapter describes the options in database backup and the utilities provided for logical backup and restoration of these databases. It also describes the procedure to change the Oracle passwords within OV Topology Server.

OV Topology Server Databases

OV Topology Server utilizes four database instances in which it maintains the various product databases. The four database instances are:

- CORBA database instance—ovcorba

  The ovcorba instance includes the following CORBA services:

  — User Service

  This service stores user login information, including login ID and which hosts and services the user may access.

  — Access Management Service (AMS)

  This service runs on top of the User Service, and stores users' operation profile information, including which applications, actions, and domains the user may access.

  — Unified Service

  This service provides the federation functionality that allows services to be accessed transparently across multiple hosts. Clients can access a registered service on any host and have access to the full distributed service provided by all collection managers on all hosts in the installation.

  The following managers are services registered with the Unified Service:

  — Alarm Manager

  — Managed Object Manager

  — Managed Object Class Manager

  — OM Event Manager

  — Outage Plan Manager

  — Problem Manager

  The following CORBA services do not use `ovcorba`:

  — Message Logging Service

  The message logging service allows collection managers to log

error, warning and informational messages. *This service does not use the ovcorba Oracle database instance.*

— Transaction Processing Service

The transaction processing service is an internal service that allows applications to group multiple steps for object updates into a single transaction. It ensures that all updates become permanent if all associated updates succeed, and that no changes are made if any associated update receives an error. *This service does not use the ovcorba Oracle database instance.*

- Notification Service database instance—ovnsls

  Information for the CORBA Notification Service is maintained in the ovnsls database instance. This service distributes notifications to GUI servers and GUI Client applications. Notifications generated indicate events such as:

  — Object Creation

  — Object Deletion

  — Object State Change

  Notifications generated may include Problem, Managed Object, Outage Plan, and/or OM Event events.

- FM server database instance—fmsdb

  fmsdb is the main database of the FM server components and resides on the FM server machine. It contains the problem database, the topology database, the outage plan database, and the OM Event database.

- Presentation Database instance—guidb

  This database instance contains the GUI presentation database and is set up on the GUIDB server machine.

CORBA

CORBA and its associated Object Request Broker (ORB) comprise a supporting technology for the integrated and distributed architecture of OVSACN.

CORBA is based on Object Management Group (OMG) standards. HP provides a central, standardized ORB, HP ORB Plus, to handle object-oriented, network-wide communications in a multi-vendor environment. For more information, see the online HP ORB Plus documentation at `/opt/orbplus/share/help/index.html`. CORBA is a technology component underlying OV Topology Server that provides services such as notification management, log message management, and distributed (unified) services management. The advantages of using CORBA include the ability to integrate a variety of software and hardware including new and legacy applications, ease of code reuse, provision of powerful tools, and ease of deployment in a distributed object environment.

Managing CORBA Hosts

OV Topology Server locations are based on CORBA hosts. A CORBA host is installed on each FM server machine, and the hosts are combined into a single distributed installation; that is, a distributed set of locations.

The Installation Management Service command-line tools allow you to obtain information about the hosts in an installation.

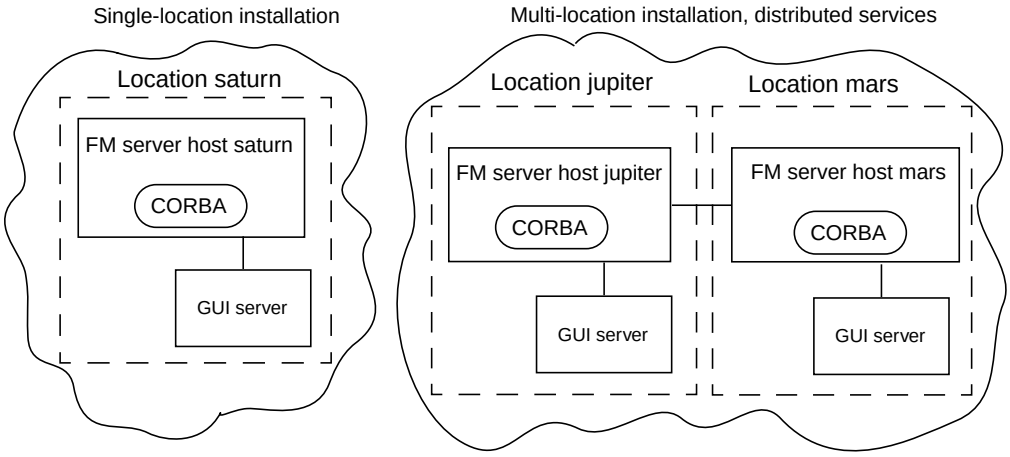
Use the installation management service command-line tools during installation. For details regarding installing OV Topology Server components, see the *HP OpenView Service Assurance for Communication Networks Installation Guide*

Overview of CORBA Installations

An installation consists of a set of hosts (FM server machines), each with CORBA installed, and all of which can be managed as a unit. Figure 3-1 illustrates two sample installations — a single location installation and a multi-host installation.

The multi-location installation has two hosts, with CORBA installed on each. Because the hosts, jupiter and mars, are in the same network and are managed as a unit, they comprise a single installation.

Figure 3-1 Two Sample Models of Installations



Each time you install CORBA on a host, that host is considered to be in its own installation. Each new host must be added to the installation to create a single distributed installation.

Only add *new, initialized* hosts to an installation. Adding an existing host with existing access management data is *not* supported because the access management data is lost in the `ovsync_amsuser` step.

Notification Service Overview

The Notification Service is the basic transport mechanism used for distribution of notifications to GUI servers or any GUI Client application. Notifications generated may include distribution of problem notifications, managed object notifications, outage plan notifications, and OM Event notifications. These notifications may indicate creation, deletion, or state change.

Managing Users

OV Topology Server uses the CORBA User Service to store user login information (for example, from which host a user may access the services). The CORBA Access Management Service runs on top of the

User Service and stores operation profile information such as which applications, actions, and domains a user may access.

NOTE

Users and user access management for the FM servers are automatically managed via the Operation Profile Configurator. Therefore, *no* CORBA-level management of these users is necessary.

However, if a GUI server is installed on a separate machine from the FM server machine, some CORBA-level user creation work is needed for the GUI server. The information is this section, “Managing CORBA Users” is typically only relevant for independent GUI servers.

Each user must have an identity that is unique within a CORBA installation. This unique identity is used to verify the user's authorization to perform operations.

Each CORBA user may be associated with one or more host-specific logins. A host-specific Unix system login is the identifier by which a user is known to the host's operating system. The association between a CORBA user and a host-specific login is examined by the User Service to determine if a user is authorized for an operation.

Overview of Unified Services

The CORBA Unified Services provide the basic mechanism for topology server collection managers to federate transparently. They hide the fact that a service, such as the Problem Service, is provided by multiple distributed instances of the same collection manager (problem manager) in different locations.

The following topology server managers are services registered with the CORBA Unified Services:

- Alarm Manager
- Managed Object Manager
- Managed Object Class Manager
- OM Event Manager
- Outage Plan Manager
- Problem Manager

Other CORBA Services are also registered with the Unified Services, for example:

- Access Management Service
- User Service
- Message Logging Service

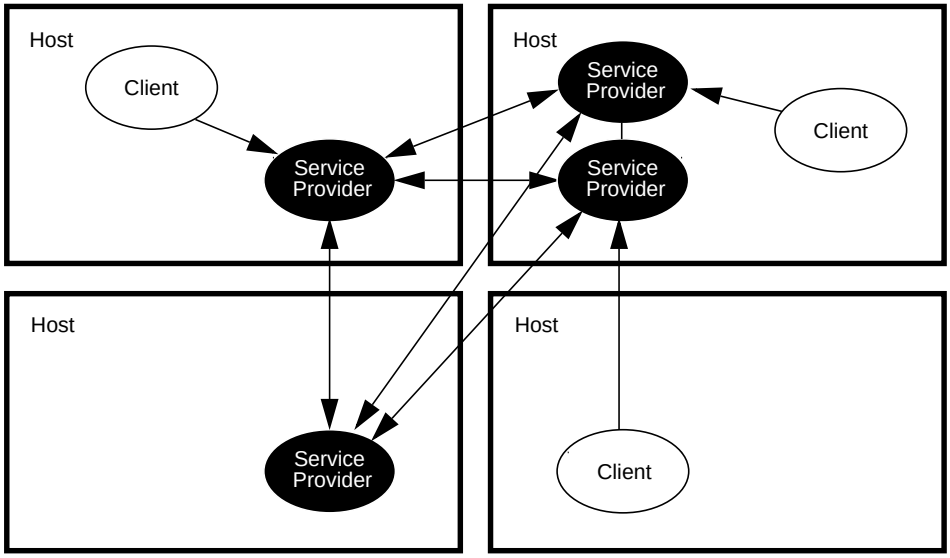
The Unified Services provide federation for registered services so the services can be distributed across multiple hosts. Clients can access a service on any host and have access to the full distributed service on all locations in the installation.

Providing a Unified Service comprised of a number of cooperating objects has the following advantages over a single object providing the same service:

- Clients do not have to know with which server object they are communicating.
- Clients do not have to know the host on which the server object resides.
- Clients still have access to the service, even if some server objects are unavailable.

Figure 3-2 shows an example of how clients communicate with the service providers that comprise a Unified Service.

Figure 3-2 The Unified Service Model



Managing CORBA Databases

CORBA stores persistent data in both the file system and the database using ODBC technology. Command-line tools that manage the databases and data files exist in: `/opt/OVCORBA/bin`, and are listed in Table 3-1.

Table 3-1 Database Management Command-Line Tools

Command-line Tool	Function
`ovcorba_db_config`	Performs database and odbc configuration for ovcorba and ovnsls services; is a link to `ovtdbadmin`.
`ovsync_amsuser`	Synchronizes the AMS and User Service information with the other nodes in an installation; needed following a hot or archive-mode physical recovery from backup.
`ov_amsuser_db`	Performs a logical backup or restore of the AMS and User Services.

These commands are discussed further within this section.

CORBA Data Directories

The locations of persistent data on each host in the CORBA installation are:

- `/var/opt/orbplus`
- `/var/opt/OVCORBA`
- `/var/opt/OVNSLS`
- `/etc/opt/OVTCOMMON`
- `/etc/opt/orbplus`
- `/etc/opt/OVCORBA`
- `/etc/opt/OVNSLS`

No access to this data is supported except via the command-line utilities provided with CORBA or documented procedures.

CAUTION Do not alter or remove any files or directories used by CORBA unless you are restoring a consistent backup to recover from a failure or you are instructed that it is safe to do so by HP product support personnel.

CORBA Database Concepts and Administration

Many CORBA Services maintain their data persistently in a Relational Database Management Server (RDBMS). The required RDBMS for CORBA is the Oracle® RDBMS product.

Oracle RDBMS

Oracle must be installed and running *before* you begin the initialization of CORBA. As CORBA is initialized, tables for CORBA Services data are created in the Oracle RDBMS.

NOTE For each run-time node, you must have Oracle running before starting the CORBA Services, so appropriate connections between server processes and the RDBMS can be established.

See the Oracle documentation for procedures on starting an Oracle instance.

Database Concepts

This section briefly describes how the CORBA Services communicate with and store data in your RDBMS. Understanding these concepts will help you as you do various administrative tasks to maintain the RDBMS. The next three subsections describe the following concepts relative to the CORBA Services:

- use of the database
- use of the database tables
- use of ODBC

Use of the Database Some CORBA Services use file systems while others use the database for storing information. Table 3-2 shows which services use which storage methods.

Table 3-2 Storage Methods Used by CORBA Services

Service	Storage Method
Message Logging	File system
Notification	Database
User	Database
Unified	Database
Access Management	Database

Use of the Database Tables As shown, some CORBA Services use special tables in an RDBMS to store their data. Additional services may use these tables in the future. You can view the schemas for these tables in the directories:

`/opt/OVCORBA/schemas`

`/opt/OVNSLS/schemas`

Database SQL files in the `schemas` directory are named based on table names as follows:

`<table name>[Create | Drop]_oracle.sql`

Use of ODBC ODBC is the mechanism which enables the CORBA Services to access your RDBMS. It is a PC and Unix-based standard that allows a single executable to converse with any number of database servers (such as Oracle, Informix, Ingres, Sybase, SqlServer, and certain proprietary databases). With ODBC, you need only one executable per service to communicate with your database.

ODBC accomplishes this by using Dynamic Link Libraries (DLLs) or shared libraries. A database client uses the ODBC API for all interactions with your Oracle database. A database-specific “driver” translates the generic ODBC function into the corresponding function(s) using the native API.

When the database client connects to a database, it does so by giving the name of a Data Source. A Data Source is a logical database name used by

ODBC to reference a particular database. The Data Source pre-configured and used by CORBA is “OpenView”. This name appears in the `/etc/opt/OVCORBA/.odbc.ini` and `/etc/opt/OVNSLS/.odbc.ini` files along with information indicating which ODBC driver (a DLL or shared library) to use and database-specific information about establishing the connection.

When the `/opt/OVCORBA/bin/ovcorba_db_config` script runs, it creates the `/etc/opt/OVCORBA/.odbc.ini` and `/etc/opt/OVNSLS/.odbc.ini` files.

Backup and Recovery

To ensure your ability to recover from a disaster, backups of key directories, the Object Locator, and the CORBA Services database should be performed regularly.

Backup of Key Directories

Back up the following directories on each Fault Management (FM) Server machine using the backup method of your choice:

- `/var/opt/orbplus`
- `/etc/opt/orbplus`
- `/var/opt/OVCORBA`
- `/var/opt/OVNSLS`
- `/etc/opt/OVCORBA`
- `/etc/opt/OVNSLS`

Backup/Recovery of the Object Locator

To create backup files for the ORBPlus Object Locator, execute:

`/opt/orbplus/sbin/sdbmcp -f /var/opt/orbplus/locator_db /tmp/locator_db`

This creates two files that must be archived:

- `/tmp/locator_db.dir`
- `/tmp/locator_db.pag`

To restore these files, use the `sdbmcp` command again, but reverse the order of the pathname parameters. For example, to restore the Object Locator, execute:

```
/opt/orbplus/sbin/sdbmcp -f /tmp/locator_db
/var/opt/orbplus/locator_db
```

After the files have been restored, verify that they have the correct permissions:

- owner = ovdb
- group = ovdb
- mode = 600

Correct the permissions if necessary.

For additional information, see the online ORBPlus documentation.

Backup/Recovery of CORBA Service Databases

Perform a physical backup of the CORBA Services Database periodically to protect against data loss. Consult your Oracle documentation for information on making physical backups. The following section describes the physical makeup of the CORBA Services Database. The sections that follow this describe issues with backup/restore that are unique to the CORBA Services Database.

Physically, an Oracle database is composed of data files, control files, online redo log files, archived redo log files (if running in archivelog mode), and configuration files (including initial parameter file `listener.ora`, and `tnsnames.ora`, etc).

There are two databases that are used by the CORBA Servers, `ovcorba` (the main CORBA Services) and `ovnsls` (the Notification Service).

The default directories for the database files (data, control, and online redo log files) are as follows:

File	Location
ovcorba data	`/var/opt/oracle/oradata/ovcorba`
ovcorba initial parameter file	`$ORACLE_BASE/admin/ovcorba/pfile`

File	Location
ovnsls data	`/var/opt/oracle/oradata/ovnsls`
ovnsls initial parameter file	`$ORACLE_BASE/admin/ovnsls/pfile`
archived redo log files	check initial parameter `log_archive_dest`
`listener.ora`, `tnsnames.ora`	`$ORACLE_HOME/network/admin`

Each database is composed of multiple tablespaces. A tablespace is composed of one or more data files.

The ovcorba database consists of the following tablespaces:

Tablespace	Purpose
`system`	stores data dictionary, etc.
`ovcorba`	stores tables used by ovcorba servers
`ovcorbaindex`	stores indexes of tables in ovcorba database
`rollback1`	stores rollback segments created for ovcorba database
`temp1`	stores temporary segments

The ovnsls database consists of the following tablespaces:

Tablespace	Purpose
`system`	stores data dictionary, etc.
`ovnsls`	stores tables used by Notification Service
`ovnslsindex`	stores indexes of tables in ovnsls database
`rollback1`	stores rollback segments created for ovnsls database
`temp1`	stores temporary segments

Types of Database Backups There are various types of database backups that can be performed:

- **Hot backups** - this is the recommended backup mode on an ongoing basis. It allows the database instances to stay up while backups and restores are taking place. For this option, the databases must be run in archive mode, thus allowing recovery to a particular point in time.
- **Cold backups** - the database instances must be shut down for backup and recovery. Cold backups can be done in either archive or non-archive mode:
  - **Archive mode** - allows recovery of a database to a particular point in time, if databases on all nodes are run in archive mode. This is strongly recommended over non-archive mode.
  - **Non-archive mode** - since automatic point-in-time recovery is not possible in this mode, all nodes must be backed up and restored at the same time. This is the least recommended option.
- **AMS and User Service backups** - regardless of which one of the three options above you choose for backing up and restoring your CORBA databases, you may need to perform this *additional* backup/restoration step for the AMS and User Service.

Performing Hot Backups Choose the node in the installation to back up, then see your Oracle documentation for details on performing the hot backup task.

Restoring from a Hot Backup When restoring from a hot backup, a restore can be performed on one node in the installation while the other nodes in the installation continue to run.

NOTE If there is not a node with good AMS and User database information, then a recovery from a logical backup must be performed as described in “Backup/Recovery of CORBA AMS and User Service Databases” on page 62.

To restore from a hot backup, perform the following on the node to be restored:

1. Restore the Oracle database from the backup as described for hot backups in your Oracle documentation.
2. Synchronize the AMS and User services information with the other nodes in the installation by running:

   ovsync_amsuser

   This command should be run after the restore. Starting the CORBA Services after running this command will not be necessary, as ovsync_amsuser will also start all of the CORBA Services on this node. The ovsync_amsuser utility chooses another host in the installation and downloads the AMS and User database information from the remote host to the host on which ovsync_amsuser is run. This requires that at least one host in the installation be up and running CORBA Services and have good AMS and User database information.

Performing Cold Archive-Mode Backups To perform a cold backup in archive mode, ensure all databases are running in archive mode, then:

1. Shut down the OVNSLS and OVCORBA Oracle instances.
2. Stop the CORBA and Notification Services using the ovcorba_admin and ovnsls_admin commands.
3. Perform the cold backup according to the instructions in your Oracle documentation.
4. Restart the Oracle instances and the CORBA and Notification Services.

Restoring from a Cold Archive-Mode Backup Restore from a cold archive-mode backup as follows:

1. Restore an individual node as described for cold archive-mode in your Oracle documentation.
2. Synchronize the AMS and User services information with the other nodes in the installation by running:

   ovsync_amsuser

   This will also automatically re-start the CORBA Services.

3. If there is not a node with good AMS and User database information, then a recovery from a logical backup must be performed as described in “Backup/Recovery of CORBA AMS and User Service Databases” on page 62.

Performing and Restoring from a Cold Non-Archive Mode This is the least recommended option. All databases must be backed up and restored simultaneously.

- To perform a backup, follow the instructions for “Performing Cold Archive-Mode Backups” on page 61 on all nodes. See your Oracle documentation for details on cold backups and non-archive mode.
- To restore from a cold non-archive mode backup, follow the instructions for “Restoring from a Cold Archive-Mode Backup” on page 61 on all nodes. All nodes should be restored with backups from the same time. See your Oracle documentation for details on cold restores.

Backup/Recovery of CORBA AMS and User Service Databases

Perform a logical backup to protect the AMS and User information from data loss. A logical backup of the User and AMS Services is performed using the utility `ov_amsuser_db`. This utility performs a logical backup and puts the files associated with the backup in a directory created and specified by the user. The utility can also perform a restore from the backup directory. The restored data is propagated to all nodes in the installation. The utility will put the User and AMS Services in read-only mode during backup and restore. As a result, no user profile data can be changed during the backup and restore operations. The `ov_amsuser_db` utility dumps the database information to a file in CORBA's CDR format. Oracle's Export/Import utilities are not used.

For example, to back up the AMS and User database information for the installation to an existing `/var/backup/100499` directory, execute this command as user `ovdb` or `root` on any host in the installation:

```
ov_amsuser_db -b /var/backup/100499
```

The following command executed from any host in the installation would restore the AMS and User database from the above described backup:

```
ov_amsuser_db -r /var/backup/100499
```

All nodes in the installation would now contain the information from the backup in the directory `/var/backup/100499`.

Time Synchronization

CORBA Services such as the Message Logging Service store time stamps when messages are generated. The time stamp is obtained from the operating system on the host where the message is logged.

For the Message Logging Service's distribution features to perform optimally, it is necessary that the system time on all hosts in the installation be kept synchronized.

Use the facilities provided by the operating system, such as the UNIX Network Time Protocol Synchronization daemon (see manpage *xntpd*(1M)) to keep the time synchronized on the hosts.

If you do not keep the system times synchronized, the results from reading a distributed log (for example, using `/opt/OVCORBA/bin/ovdumplog`) will contain messages in a time sequence that is different from the one in which they actually occurred.

Managing CORBA Users

CORBA User Service stores user login information, from which the GUI server machine can allow a user access to the OV Topology Server. The CORBA Access Management Service (AMS) runs on top of the User Service and stores operation profile information such as which applications, actions, and domains a user can access.

NOTE Users and user access management for the FM servers are automatically managed via the Operation Profile Configurator, thus *no* CORBA-level management of these users is necessary.

Each user must have a unique identity within an installation that is used to verify the user's authorization to perform operations.

Each user may be associated with one or more host-specific logins. A host-specific UNIX system login is the identifier by which a user is known to the host's operating system. The association between a user and a host-specific login is examined by the User Service to determine if a user is authorized for an operation.

These following command-line tools in `/opt/OVCORBA/bin` are provided for CORBA-level user management:

Table 3-3 User Management Service Command-Line Tools

Command-line Tool	Function
`ovuser`	Add, remove, and list CORBA users.
`ovlogin`	Add, remove and list host-specific logins for specified CORBA users.

Examples follow in this section. See the online Reference Pages manual for more details about these commands.

NOTE These commands are not needed to create users and must not be used as such. Their main purpose would be to list the users on specific servers/locations.

Listing CORBA Users

Because the operation profile configurator handles the set up of CORBA users for FM servers, there is not typically a need to manage the users from CORBA.

However, if you would like to list current valid CORBA users in the installation from the command line, execute:

```
/opt/OVCORBA/bin/ovuser -l
```

This returns a list of *all* CORBA users in this installation. You can use command-line options to restrict the list to specific users.

For additional information on the `ovuser` command and other user management options, see the online Reference Pages manual.

Adding, Removing, and Listing Host-Specific Logins

You can add host-specific logins, remove host-specific logins, and obtain a list of current HP OVC CORBA users and their host-specific logins using the `/opt/OVCORBA/bin/ovlogin` command-line tool.

For example, to create a host-specific login of “Joe_Smith” for a CORBA user called “Joe” on host `jupiter`, execute the following on jupiter:

```
/opt/OVCORBA/bin/ovlogin -a Joe Joe_Smith
```

To accomplish the same task working remotely from `jupiter`:

```
/opt/OVCORBA/bin/ovlogin -a Joe Joe_Smith@jupiter
```

To remove all logins for CORBA user “Joe”, execute:

```
/opt/OVCORBA/bin/ovlogin -r Joe alllogins
```

To list all the CORBA users and host-specific logins in this installation, execute:

```
/opt/OVCORBA/bin/ovlogin
```

You can use command-line options to restrict the list to specific users or specific hosts.

To add two administrator logins on a stand-alone GUI server machine, execute:

ovlogin -a oemfadm oemfadm@*guis_host*

ovlogin -a OVCORBA_Administrator root@*guis_host*

where *guis_host* is one of the following:

- the *unqualified* name of the GUI server machine if the GUI server machine is in the *same* network domain as its FM server machine.
- the *fully qualified* name of the GUI server machine if the GUI server machine is in a *different* network domain from its FM server machine.

Backup/Recovery of FM Server Databases

OV Topology Server stores information in the following databases on the FM servers:

- Problem database

  This database contains all details of the problems received, such as the problem ID, the details of the managed object that sent the problem, the date and time of receipt of the problem, and the subsequent update time.

- Outage Plan database

  This database contains details on the outage plans created and the owner of the plan.

- OM Event database

  The OM Event database stores the event details including those of the operator performing an operation on the event such as owning or discharging the event.

- Topology database

  This database contains details of the managed object instances.

- SDR database

  This database contains the details of the system distribution rules. It contains details of the locations and partitions in the OV Topology Server site.

All the data processed by the FM servers are housed in the Oracle database instance—`fmsdb`. The database backup and recovery strategy addresses the protection of data from loss by system or media failure. The strategy can be divided into two parts: physical backup and recovery and logical backup and recovery.

Physical Backup/Recovery

Physically, the Oracle database is composed of datafiles, controlfiles, online redo log files, archived redo log files (if running in archivelog mode) and configuration files (including initial parameter file, `listener.ora` and `tnsnames.ora`). The default directories for these files are as follows:

Datafiles, controlfiles, online redo log files:

`/var/opt/oracle/oradata/fmsdb`

Archived redo log files:

directory defined by the parameter `log_archive_dest`

Initial parameter files:

`$ORACLE_HOME/dbs/initfmsdb.ora`

Configuration files
listerner.ora, tnsnames.ora:

`$ORACLE_HOME/network/admin`

Usually, the configuration files will not change much and you can keep a record of the file on paper or in backup files.

All other files must be backed up periodically.

In the case of a media or a system failure, the backup strategy you adopt dictates the recovery method available. There are two kinds of physical backups: cold backup and hot backup.

Cold Backup

To take cold backups, the `fmsdb` instance must be shut down. Then using the operating system commands, you can copy the files to disks or tapes. The datafiles, control files and online redo files must be backed up.

If you run the database instance in archivelog mode, you must back up the archived redo files as well. After you have performed a cold backup, the archived redo log files created prior to the current backup become obsolete and can be removed. However, it is recommended that you maintain two cold backups and all the current archived redo log files.

If the `fmsdb` instance is running in noarchivelog mode (which means there are no archived redo files), then the cold backup is composed of data files, controlfiles and online redo logfiles only.

Hot Backup

Hot backup is available only when `fmsdb` instance is running in archivelog mode. In a 24x7 production environment, hot backup is the only form of physical backup available.

The logical unit of hot backup is the tablespace. The procedure for hot backup is:

1. Set tablespace in hot backup state.
2. Execute operating system level commands to backup the datafiles that compose the tablespace being backed up.
3. Set the tablespace back in normal state.

The redo files created during the hot backup must be backed up as well as the system requires them during the recovery process.

It is recommended that a schedule be drawn up before performing any hot backup.

For example, the following schedule may be used:

- Backup tablespace for `fms_data` on Monday
- Backup tablespace for `fms_index` on Tuesday
- Backup tablespace for `fmstopo_data` on Wednesday
- Backup tablespace for `fmstopo_index` on Thursday
- Backup tablespace system on Friday
- Backup other tablespaces on Saturday

The backup of all the tablespaces with the redo logfiles generated during the backup form the valid physical backup of the whole database instance.

Recovery

The recovery implication of physical backup can be described briefly in the Table 3-4.

Table 3-4 Recovery Implications

	Cold Backup (archive mode)	Cold Backup (non-archive mode)	Hot Backup (archive mode)
Instance failure	Recoverable	Recoverable	Recoverable
Media failure	Recoverable	Recoverable	Recoverable
User error[a]	Recoverable with limitations[b]	Not recoverable	Recoverable with limitations

a. Accidently dropped table
b. With a cold backup taken at time t1, the user accidently dropping the table at time t2, and the error noticed and the `fmsdb` instance shut down at time t3. To recover the table, replace the database with the backup taken at t1. Then apply the redo logfiles just preceding t2. Startup the `fmsdb` instance. The changes made between t2 and t3 are lost.

Database Housekeeping at Application Level

Utilities can be run to back up and restore the records in the following databases at application level:

- Problem database
- OMEvent database
- Outage Plan database

These utilities must be run by the `oemfadm` user on the FM server on which the database must be backed up or restored. This section explains the procedure to use these utilities.

Using FM Server Backup/Restore Utilities

To backup the FM server, the Oracle backup rollback segment must be offline. However, for the backup and restore utilities, the rollback segments must be online. Hence, the procedure to run any of these utilities is as follows:

1. Log on as `oracle` on the FM server on which you wish to backup or restore a database.

2. Set environment variables ORACLE_HOME, ORACLE_SID and SHLIB_PATH. Enter:

   export ORACLE_HOME=/opt/app/oracle/product/8.0.6

   export ORACLE_SID=fmsdb

   export SHLIB_PATH=$SHLIB_PATH:$ORACLE_HOME/lib

3. Use svrmgrl to put the rollback segment online. Enter:

   a. **svrmgrl**

   b. **connect internal**

   c. **alter rollback segment rollb_oemf_batch online**

   d. **exit**

4. Log on as oemfadm.

5. Run the appropriate backup or restore utilities as described in the following sections.

6. Log on as oracle on the FM server.

7. Set environment variables ORACLE_HOME, ORACLE_SID, and SHLIB_PATH. Enter:

   export ORACLE_HOME=/opt/app/oracle/product/8.0.6

   export ORACLE_SID=fmsdb

   export SHLIB_PATH=$SHLIB_PATH:$ORACLE_HOME/lib

8. Use svrmgrl to put the rollback segment offline. Enter:

   a. **svrmgrl**

   b. **connect internal**

   c. **alter rollback segment rollb_oemf_batch offline**

   d. **exit**

Note on Date Specification For all database backup and restore commands, the period for which the backup/restore must be executed is specified. The parameters used are startdate-time and enddate-time.

The format for date specification in all the backup utilities is as follows:

dd-mm-yyyy-hh24:mi:ss

where *dd* stands for the date within the month. The *mm* is the month. The

yyyy stands for the year of the record. *hh24:mi:ss* is the hour, minute, and second of the day from which the backup must be started. (This is in the 24 hour time format.)

The startdate-time is optional. If you do not state the start date and time, the system takes the date of the earliest record in the database as the start date, and the time as 00:00:00 (midnight of the start date).

The enddate-time is optional. If you do not specify the end date and time, the system backs up records until midnight of the date preceding the current date for the backup utilities.

For the restore utilities, the date format to be used is:

dd-mm-yyyy

The startdate is optional. If you do not state the start date and time, the system restores records from the earliest date in the backup directory.

The enddate parameter is optional for the restore utilities, too. If you do not state the end date, the system will restore records until midnight of the date preceding the current date. The date is entered as indicated for the `startdate` parameter.

NOTE

If the start date and end date are for the same date, then the database is backed up for the specified time of the given date. However, if the two dates are different, then the system backs up from 00:00:00 (midnight) of the start date specified to 00:00:00 (midnight) of the date following the end date specified.

For example, if the duration for backup is specified as:

```
startdate-time 15-02-2000-11:00:00
enddate-time    15-02-2000-19:00:00
```

The database records will be archived from 11 a.m. to 7 p.m. for the 15th of February 2000.

If the following duration is specified:

```
startdate-time 15-02-2000-11:00:00
enddate-time    16-02-2000-19:00:00
```

The database records will be backed up from:

```
startdate-time 15-02-2000-00:00:00
enddate-time    17-02-2000-00:00:00
```

Backing Up FM Server Databases at Application Level

Information contained in the problem, outage plan and OM event databases can be backed up using the following utilities. (Topology management is described in the next chapter.)

`fmsalmbackup`	To back up the problems database.
`fmsoutagebackup`	To back up the outage plan audit trail.
`fmsomeventbackup`	To back up the OM Event audit trail.

CAUTION

Do not use the same directory for `fmsalmbackup`, `fmsoutagebackup`, or `fmsomeventbackup`. If all backups are to the same directory, the recovery process may restore incorrect data.

This section explains these utilities all of which must be run only by the `oemfadm` user and apply to the database on the logged in FM server, only.

NOTE

These utilities are intended to be used only for backing up data from and restoring data to the same machine. They are not a mechanism for moving data to another machine in the installation. Each backup file contains information specific to the machine from which it was taken.

Backing Up Problem Database Alarms received by an FM server are stored in the problem database. As problems are received, they are displayed in the problems presenter, and stored in the problem database. When the problems are resolved, they are discharged from the problem table. These problems can be taken out of the problem database to free disk space.

The problem backup utility, `fmsalmbackup`, is used to back up problems that have been resolved and discharged from the Problems Presenter *as well as unresolved problems*. The problems can be backed up for any specific period of time.

This utility can be executed online while the FM server is running. Because this utility accesses the problem database, it affects the performance of the server. Hence, it is recommended that you run the utility when the problem traffic is the lightest, and that you run it

periodically (daily, bi-weekly, or weekly) depending on the problem traffic. The command must be run by the `oemfadm` user. It backs up alarms from only the logged in FM server. The backed up problems can be restored if required.

Run the following command as step 5 of “Using FM Server Backup/Restore Utilities” on page 70:

`fmsalmbackup`

You will be prompted for confirmation before the command is executed. Confirm the date and the directory for backup.

If you have specified the -u parameter, then you will be prompted to confirm that the local FM server and all GUI servers have been shut down.

Informational messages are displayed on the screen as problems are backed up for each date indicated in the `fmsalmbackup` command. Statistics of the number of records backed up for each date, in each of the files, are also displayed for your information.

See “Recovering the Problem Database” on page 35 for details on restoring the problem database.

See “Recovering the Problem Database” on page 35, for details on recovering the problem database after an abnormal system shutdown, or an inconsistent database.

Backing Up Outage Plan The outage plans created by the operators are stored in the outage plan database. This database contains all the records relating to outage plan creation, modification, and object restoration. It also stores audit trails of activities involved in each object going on outage.

The `fmsoutagebackup` utility is used to archive the outage plan audit trail. It can be run online while the FM server is running. It is recommended that you run the utility when the problem traffic is lightest, and that you run it periodically (daily, bi-weekly, or weekly) depending on the number of outage plans. Clearing the outage plan database of expired plans frees diskspace and helps hold down the database size.

Run the following command as step 5 of “Using FM Server Backup/Restore Utilities” on page 70:

```
fmsoutagebackup
```

You will be prompted for confirmation before the command is executed. Confirm the date and the directory of the backup.

Statistics of the records backed up are displayed on the screen. The records corresponding to the backed up audit trails are removed from the outage plan database files.

See “Restoring the Outage Plan Database” on page 77, for details on restoring the backed up outage plan audit trails.

Backing Up OM Event Database The OM Events and their processing information are stored in the OM Event Database. This database contains all the records related to OM Event modification and deletion of managed object events and processing of these OM Events. It stores the audit trail of activities for each OM Event.

The `fmsomeventbackup` utility is used to archive the audit trails of the OM Events. It can be executed while the FM server is running. It is recommended that you run the utility when the OM Event traffic is lightest, and that you run it periodically (daily, bi-weekly, or weekly) depending on the volume of OM Events. Clearing the OM Event database of discharged events frees diskspace and maintains the database size.

Run the following command as step 5 of “Using FM Server Backup/Restore Utilities” on page 70:

```
fmsomeventbackup
```

Informational messages are displayed on the screen as the OM Event audit trails are backed up. Statistics on the records backed up for each date, in each of the files, are also displayed for your information. Finally, the records pertaining to the audit trails that had been backed up are removed from the OMEvent Database files.

See “Restoring the OM Event Database” on page 77, for details on restoring the backed up information.

Restoring Application Level Backup

This section describes the procedure to restore the databases archived as described in the section “Backup/Recovery of FM Server Databases” on page 67. All these utilities must be run as user `oemfadm` on the FM server on which the details must be restored.

These utilities use the Oracle utilities export and import for restoration and write into the corresponding database. Before running these commands verify that:

- The login path includes the directory in which the Oracle utilities are stored.
- Oracle is running when the command is being executed.

NOTE See the Note on Date Specification, on page 71, for information on specifying dates for the utilities described in the following sections.

NOTE These utilities are intended to be used only for backing up data from and restoring data to the same machine. They are not a mechanism for moving data to another machine in the installation. Each backup file contains information specific to the machine from which it was taken.

Restoring the Problem Database The `fmsalmrestore` command is used to restore the problems backed up using the `fmsalmbackup` command. This command can be run while the FM server is running. It is recommended that after the restored problems have served their purpose, they are backed up and deleted from the database using the `fmsalmbackup` command.

Run the following command as step 5 of "Using FM Server Backup/Restore Utilities" on page 70:

`fmsalmrestore`

You will be prompted for confirmation before the command is executed. Confirm the date and the directory for restoration.

For the records are being restored, informational messages are displayed on the screen as problems are restored from each of the backup files, for each date indicated in the `fmsalmrestore` command.

When you restore problems, the utility may encounter cases of duplicate problems. That is, some of the problems that you are trying to restore may already be present in the database, due to the restoration of the same problems more than once. In that case, the utility will prompt you to confirm overwriting the existing problems. If you confirm this, it will

delete the records in the database and restore them from the backup. If you select not to overwrite the records in the database, restoration for that date will be aborted, and you will be prompted to confirm that the command continue operations.

Restoring the Outage Plan Database The `fmsoutagerestore` command is used to restore the outage plan information backed up using the `fmsoutagebackup` command.

Run the following command as step 5 of "Using FM Server Backup/Restore Utilities" on page 70:

`fmsoutagerestore`

You will be prompted for confirmation before the command is executed. Confirm the date and the directory for restoration.

If the records are being restored in the database, informational messages are displayed on the screen as the records are restored from each of the backup files, for each date indicated in the `fmsoutagerestore` command.

When you restore outage plan audit trails, the utility may encounter cases of duplicate records. That is, some of the audit trails that you are trying to restore may already be present in the database, due to the restoration of the same records more than once. In that case, the utility will prompt you to confirm restoring the duplicate records. If you confirm this, it will restore them from the backup. Otherwise, the duplicate records will not be restored.

If the command fails during execution, you can run the command again and it will detect the point of failure and re-start from that point.

Restoring the OM Event Database The `fmsomeventrestore` command is used to restore the OM Event records that were backed up using the `fmsomeventbackup` command.

Run the following command as step 5 of "Using FM Server Backup/Restore Utilities" on page 70:

`fmsomeventrestore`

You will be prompted for confirmation before the command is executed. Confirm the date and the directory for restoration.

If records are being restored in the database, then informational messages are displayed on the screen as event messages are restored from each of the backup files, for each date indicated in the `fmsomeventrestore` command.

Note that when you restore OM Event records, the utility may encounter cases of duplicate records. That is, some of the records that you are trying to restore may already be present in the database, due to the restoration of the same records more than once. In that case, the utility will prompt you to confirm restoring the duplicate records. If you confirm this, it will restore them from the backup. Otherwise, the duplicate records will not be restored.

If the command fails during execution, you can run the command again and it will detect the point of failure and re-start from that point.

Maintaining the Presentation Database

Display information applicable to the GUI Client, such as object symbol details, the application window description, user settings, and user-specific maps, are stored in the presentation database, also referred to as GUIDB. This information is initially configured and loaded using the `guidbsysconfig` utility, which is run after the GUIDB is installed.

The following utilities have been provided to maintain the GUIDB:

`guidbbkup` to backup the presentation database.

`guidbrestore` to restore presentation database information from a backed up file.

These two utilities are described in this section.

Backing up the GUIDB

The GUIDB backup command must be run on the server on which the presentation database has been installed. It must be run under the `root` or `oemfadm` login. The command to back up the presentation information stored in the GUIDB is:

`guidbbkup`

Restoring the GUIDB

GUIDB information, which is backed up using the `guidbbkup` utility, can be restored using the `guidbrestore` utility. The `guidbrestore` utility restores the GUIDB to the state it was when the backup was taken.

The server on which the GUIDB has been installed must not have any OV Topology Server processes running when this database is being restored. Before running the command to restore the GUIDB, ensure that Oracle is running.

To restore the GUIDB, log in as `root` or `oemfadm` and execute:

`guidbrestore`

This command restores the GUIDB from the specified file. Note that this command overwrites the existing information. This could result in loss of data from the time of the restore file date to the current date.

If you do not wish to lose the changes made during the period between the database backup and its restore, you could take another backup at the current time before restoring the old GUIDB.

NOTE Before starting the GUI server, also restore the AMS database. For information on this process, see “Backup/Recovery of CORBA AMS and User Service Databases” on page 62.

When the GUI servers are started up after the GUIDB has been successfully restored, and the topology GUI presenters opened, the presenters are displayed according to the current GUIDB information from the restored database.

Physical Backup of GUIDB

When backing up the GUIDB in a replicating environment, it is recommended to back up the entire Oracle SID. This may be done through methods recommended by Oracle. Oracle Backup Manager available through Oracle Enterprise Manager is the recommended tool for backing up the GUIDB.

The GUIDB is to remain highly available so that the database is converted from `NOARCHIVELOG` to `ARCHIVELOG`. This allows greater flexibility in backing up the database while the system is online. To perform the conversion, log off all users. Then log on as user `oracle` and run `svrmgrl` on the server machine:

```
# svrmgrl
SVRMGRL> shutdown
SVRMGRL> startup mount exclusive
```

```
SVRMGRL> alter database archivelog;
SVRMGRL> shutdown
SVRMGRL> startup
```

The most active table spaces are `GUIDB_TABLES` and `GUIDB_INDEXES`. See the Oracle Enterprise Manager for details.

NOTE The script `guidbbkup` only saves the GUI related data. It does not save critical replication environment information. If this tool is used, the re-configuration of the database and replication is required before restoring the data.

OV Topology Server Backup Schedule

Table 3-5 provides a list of suggested backup schedule to enable a comprehensive system backup for installation.

Table 3-5 Database Backup Suggestions

Dirs/Files/DBs to back up	Backup utility/procedure	Location/When
Physical backup of: fmsdb	Use Oracle online backup.	Location: FM servers When: biweekly (used to restore tables if tables are damaged in Oracle)
Logical backup of a specified set of: Alarms OM Events (audit trail) Outages (audit trail) Managed Object Instances	fmsalmbackup fmsomeventbackup fmsoutagebackup fmsmhimport	Location: FM servers When: daily
Physical backup of: guidb	Use Oracle online backup.	Location: GUIDB When: after every major change to the GUIDB. For example, after installation, and after any substantial change to the topology database or operator configuration. Also, periodically save changes to user and administration settings and maps.
Logical backup of: guidb	guidbbkup	Location: GUIDB When: Not the recommended way to backup the GUIDB; instead, use this file for support questions and troubleshooting.

Table 3-5 Database Backup Suggestions

Dirs/Files/DBs to back up	Backup utility/procedure	Location/When
CORBA data directories: • `/var/opt/orbplus` • `/etc/opt/orbplus` • `/var/opt/OVCORBA` • `/var/opt/OVNSLS` • `/etc/opt/OVCORBA` • `/etc/opt/OVNSLS`	Use any desired method to copy these directories to a backup location (can use the `tar` utility) FM server can be running.	Location: FM servers When: following software installation or any initialization (`ovcorba_admin -init`, `ovnsls_admin -init`, `fmsinit`, etc.)
`/var/opt/OVCORBA/msglog_server`	Use any desired method to copy these directories to a backup location (can use the `tar` utility) FM server can be running.	Location: FM server When: as often as you want the error log saved
Object Locator database	Execute `/opt/orbplus/sbin/sdbmcp` then copy the resulting `.dir` and `.pag` files to a backup location FM server can be running.	Location: FM server When: following software installation or any initialization (`ovcorba_admin -init`, `ovnsls_admin -init`, `fmsinit`, etc.)
• OVCORBA databases (User, Unified, and Access Management Services) • OVNSLS Notification Service database	Use either Oracle's hot physical backup procedure or cold physical archive-mode procedure: *Hot physical backup*: can be performed while all CORBA Services and the FM server are up *Cold physical backup*: must bring down FM server and CORBA Services before doing backup or restore	Location: FM servers When - OVCORBA: following software installation, any initialization, or any change in User/AMS configuration or operator configuration. When - OVNSLS: schedule on a regular basis; data will change as various operation profiles with various filters log on

Table 3-5 Database Backup Suggestions

Dirs/Files/DBs to back up	Backup utility/procedure	Location/When
AMS and User Services logical data	To perform a logical backup of this data: 1. Create a directory into which the data will be saved 2. Execute the `ov_amsuser_db -b` command 3. Copy the directory to a backup location FM server must be running.	Location: any FM server in the system (restore automatically replicates the data to all other servers) When: following the physical backup of the AMS and User CORBA databases

Changing Oracle User Passwords within OV Topology Server

The Oracle users `fmsadm` and `fmsguest` are created specifically for OV Topology Server. These user IDs are used by the FM server to access the various databases. The passwords for these users are set during FM server configuration using `fmssysconfig`.

The passwords are encrypted and stored in the following files under the `$FMSETC/share/conf` directory. *These files must not be edited manually.*

`fmsdbauth.conf`	for the `fmsadm` user
`fmsdbgauth.conf`	for the `fmsguest` user

For security reasons, it is recommended that the passwords of these users, `fmsadm` and `fmsguest`, be changed periodically.

CAUTION

In the location where the Oracle user passwords are being changed, the FM server must be shut down before changing the passwords.

The Oracle passwords for `fmsadm` and `fmsguest` must be the same at all locations.

Changing these passwords is a four-step process:

1. Log on as `root` and shut down the FM server.
2. Change the password within Oracle using Oracle commands.

   For details, see the Oracle administration manuals.
3. Change the password within OV Topology Server by:

   a. Changing the password on the FM server using the utility `oemfpasswd`.

      This utility is described in “Using oemfpasswd” on page 86
4. Start up the FM server.

Using oemfpasswd

The oemfpasswd utility is used to encrypt and update the OV Topology Server Oracle-user password on the FM server machine. It can be run only under the oemfadm login ID

To change the password of the Oracle user within OV Topology Server:

1. Log on as oemfadm on the FM server
2. Run the password change utility as follows:

   /opt/OEMF/V5.0/common/sbin/oemfpasswd -m*username* -p*password*

   where:

username	is the name of the Oracle user whose password you have changed. This must be fmsadm or fmsguest.
password	is the new password for the username entered with the -m parameter.

   For example, to change the password of the fmsguest to newone2, enter the command as follows, as soon as you have changed the password within Oracle:

   oemfpasswd -mfmsguest -pnewone2

   This command updates the $FMSETC/share/conf/fmsgauth.conf file with the encrypted password.
3. Start up the FM server.

4 Managing a Project

This chapter describes how to:

- Modify XML configuration files to the project.
- Validate the project.
- Interpret the configuration data stored in the project.
- Distribute the configuration data to the appropriate systems.
- Reapply a project.
- Restore backup configuration files.

Introduction

The concept of a project is central to the solution configuration process. All XML configuration files for a managed network solution are stored in a project directory and are listed in a *Project*.xml configuration file. Only those configuration files defined in a project can be interpreted and distributed to the OVO server and OV Topology Server.

The project directory is populated with XML configuration files created in its `XML` subdirectory, including:

- `TopoModel.xml`, which contains the definitions to the managed object classes needed to configure a telecom network. Only one TopoModel.xml file should be defined per project.
- `TopoData.xml`, which contains the managed object instance definitions. Only one TopoData.xml file should be defined per project.
- `Mappings.xml`, which contains mapping information needed to connect OV Topology Server to OVO. Only one Mappings.xml file should be defined per project.
- An `agents` subdirectory containing one `Agent.xml` file. Multiple agent XML configuration files are allowed. One agent XML file should be defined per agent installed.

Validating a Project

A set of DTD files that describe the syntax required for the XML configuration files are installed with this product. These DTD files cannot be edited, and are stored in `/etc/opt/OV/share/conf/Telco/dtds`.

Each XML configuration file has an associated DTD file that defines its acceptable syntax. It is important to validate your XML configuration files against the associated DTD files to make sure they adhere to the predefined structure before distributing your project configuration data.

To validate your project, execute:

`ovcfgvalidate <project>`

This command validates each XML file defined in *Project*.xml. First, global configuration data stored in TopoModel.xml, TopoData.xml, and Mappings.xml are validated, then each agent configuration file is validated against the global configuration data.

All validation errors appear on screen. A record is stored in `/var/opt/OV/share/log/Telco/configurator.log`.

NOTE This command is optional in the sense that when the deploy command is called, all configuration files identified in the *Project.xml* file are again validated.

Deploying a Project

A network administrator can validate the XML configuration files and generate the target configuration files using the ovcfgdeploy command.

To generate target configuration files from the XML configuration files, execute:

```
ovcfgdeploy -a [-force] <project>
```

ovcfgdeploy converts the XML configuration data into separate configuration data files needed by OVO and OV Topology Server. See its man page for more details.

The -force option generates all new target configuration files regardless of whether the deploy should be an update or not.

The ovcfgdeploy command places files in three subdirectories of the project directory:

- generated, where the target files are stored.
- backup, where copies of the XML configuration files are stored. This directory contains copies of project configuration files at the time when ovcfgdeploy is run.

  A hidden file is also created in the backup directory that is read by the ovcfgdeploy command. This file tells whether the configuration data is new or needs only to be updated.
- backup*timestamp*, where a copy of the backup XML configuration files are stored. This directory contains copies of project configuration files that were located in /backup at the time when ovcfgdeploy is run. The timestamp reflects the time of the last backup, and not the time the last ovcfgdeploy is run.

Applying a Project

After a project is deployed, run the required apply commands on the target systems to copy and install the project configuration data to their proper destination.

Four apply commands are provided to pull the configuration files to the OVO server and the topology server:

- **On the topology server:**
  `ovtopomodel.apply <project>`
- **On the topology server:**
  `ovtopodata.apply <project>`
- **On the OVO server:**
  `ovoconf.apply <project>`
- **On the OVO server:**
  `ovagt.apply <project>`

NOTE These commands must be run in the order listed above.

Table 4-1 lists each of the apply commands, what the commands do, and on which server the commands must be run.

Table 4-1 Description of Apply Commands

Apply Command	Location to Run Command	Description
`ovtopomodel.apply`	Topology server	Pulls configuration files related to the network object model from the OVO server to the topology server.
`ovtopodata.apply`	Topology server	Pulls configuration files containing the definitions of object instances from the OVO server to the topology server.

Table 4-1 Description of Apply Commands

Apply Command	Location to Run Command	Description
`ovoconf.apply`	OVO server	Adds modifications to the node bank and node assignments defined by the Mappings.xml file. Also notifies the telecom adapter of any new mappings.
`ovagt.apply`	OVO server	Pulls the agent configuration files from the OVO server to the appropriate file placement on the OVO server.

See their man pages for more details.

NOTE The apply commands detect if configuration files have not been updated since the last apply, and exits if no changes must be applied.

Removing Topology Data

Use the command `ovtoposrv.clean` to remove all topology data from the topology server system. See its man page for more details.

It is normally used after `ovcfgdeploy` has deployed the new configuration data to the topology server and an administrator has applied the data to the topology server under one of the following conditions:

- A new project is started.
- MOCs are deployed in a non-additive mode resulting in some topology data not being valid.
- A distribution rule is modified in a non-additive mode resulting in topology data not being valid.
- A new topology server built up from scratch.

Redeploying a Project

If for some reason the configuration process does not apply as expected, follow these steps to reapply the configuration data:

- Use the command `ovtoposrv.clean` to remove all topology data from the topology server system.
- Execute the following command to generate target configuration files from the XML configuration files:

  `ovcfgdeploy -a -force <project>`
- Execute the following commands to apply the generated configuration files to the target systems:

  — **On the topology server:**
  `ovtopomodel.apply <project>`

  — **On the topology server:**
  `ovtopodata.apply <project>`

  — **On the OVO server:**
  `ovoconf.apply <project>`

  — **On the OVO server:**
  `ovagt.apply <project>`

Restoring Backup Configuration Files

To restore configuration files from a backup version, follow these steps:

- Copy all configuration files from either the `backup/` directory or a `backup`*`timestamp`*`/` directory to the `XML/` directory under the project directory.
- Use the command `ovtoposrv.clean` to remove all topology data from the topology server system. This is so new configuration files are generated when the deploy command is run.
- Execute the following command to generate target configuration files from the XML configuration files:

  `ovcfgdeploy -a <`*`project`*`>`
- Execute the following commands to apply the generated configuration files to the target systems:

  — **On the topology server:**
  `ovtopomodel.apply <project>`

  — **On the topology server:**
  `ovtopodata.apply <project>`

  — **On the OVO server:**
  `ovoconf.apply <project>`

  — **On the OVO server:**
  `ovagt.apply <project>`

Removing Backup Configuration Files

Each time the `ovcfgdeploy` command is run, a backup directory containing all of the project XML configuration files is created. It is up to the administrator to decide which backups to save and which to remove. The administrator may want to automate this process by saving the backup directories monthly to a disk.

5 Managing Topology Objects

This chapter describes how to maintain a configured network. In particular, how to add, modify, and remove managed object classes, managed object instances, agent data, OM events, and objects in topology maps.

Managing Object Classes

This section highlights some the common tasks associated with managing topology object classes. In particular, adding and removing managed object classes.

All network object model definitions are stored in TopoModel.xml. This XML file contains the definitions to the managed object classes needed to configure a telecom network.

Adding Managed Object Classes

To add a managed object class to TopoModel.xml, execute:

`ovcfgtopomodel -addMoc <MOC name> <oid> <equipment type> <alternate parents> <naming attribute> <xml_file>`

Table 5-1 provides a detailed description of the arguments of ovcfgtopomodel.

Table 5-1 Adding MOCs Table

Parameter	Description
Managed Object Class Name	This is the unique name that distinguishes the object class. Enter an alphanumeric name (up to 32 characters) with initial alpha character. Hyphens and underscores are allowed. The class name `root` is reserved for internal use and cannot be used.
Object Identifier (OID)	This is the object identifier (OID) – the registration identity assigned to the object class. This is a global identifier written in dot notation. For example: `1.2.3.2.` The value begins with the number 0, 1 or 2 and ends with a digit. The numbers must not include leading zeros. The second subidentifier must be less than 40.
Equipment Type	This defines the managed object class type. Chose one of the enumerated types: **NW** (network), **NE** (network element), **TP** (termination point), **CX** (connection), or **CP** (component).

Table 5-1 Adding MOCs Table

Parameter	Description
Naming Attribute	Each object is linked to one naming attribute that uniquely identifies an attribute for the object type. The alphanumeric name for the attribute with an initial alpha character can have a maximum of 32 characters. Hyphens and underscores are allowed. By default, if no naming attribute is assigned to an object class, then `name-id` is assigned, where name is the name of the managed object class. The naming attribute must be unique, and its details must be entered in the Naming Attributes table.
Alternate Parents	This is the object class name of the parent object class. It is the name of the object class that contains the object class being defined. The alternate parent must have a corresponding record in this table.
XML file	The name of the XML configuration file to which you are adding a managd object class definition.

NOTE Be aware that after managed object class information has been entered, you must make changes cautiously to them as there may be ramifications. For example, deleting a network element class causes all components, termination points, and connections defined under this network element class to become invalid.

Removing Managed Object Classes

To remove a managed object class from TopoModel.xml, execute:

`ovcfgtopomodel -remMoc <MOC name> <xml_file>`

Table 5-2 provides a detailed description of the arguments of ovcfgtopomodel.

Table 5-2 Removing MOCs Table

Parameter	Description
Managed Object Class Name	This is the unique name that distinguishes the object class. Enter an alphanumeric name (up to 32 characters) with initial alpha character. Hyphens and underscores are allowed. The class name `root` is reserved for internal use and cannot be used.
XML file	The name of the XML configuration file from which you are removing a managd object class definition.

Updating Configuration with New Topology Model

After managed object class definitions are added or removed, TopoModel.xml must be validated, deployed, and applied to the topology server before the new configuration data is functional.

- After entering the network object model data in TopoModel.xml, validate and deploy the XML file. The validation process checks to see if the content you entered is consistent and valid against `TopoModel.dtd`. The deployment process generates target configuration files and stores them in `generated/` under the project directory.

  Execute: `ovcfgdeploy -topomodel <project>`

  If TopoModel.xml is valid, then the command generates the following configuration files: `fmpmap.conf`, `ocinfo.conf`, and `oid.conf`.

- To pull the topology model configuration files stored on the OVO server to the topology server for processing by OV Topology Server, use the `ovtopomodel.apply` command.

  On the topology server, execute:

  ```
  ovtopomodel.apply <project>
  ```

  Answer `Yes` when prompted whether to synchronize the system distribution rules and update your configuration data on the topology server.

Managing Topology Instances

All upper topology instance definitions are stored in TopoData.xml. This XML file contains the definitions to the managed object instances needed to configure the upper topology of a telecom network.

This chapter describes the utilities and procedures used to:

- Add or remove managed object instances in the topology database and their corresponding symbols in the presentation database.
- Backup managed object information from the topology and presentation databases by exporting this information to an ASCII text file.
- Troubleshoot topology database inconsistencies.

Adding Managed Object Instances

- To add a network managed object instance to TopoData.xml, execute:

  `ovcfgtopodata -addInst <parent FDN> <MOC name> <RDN> <shortname> [domain] <xml_file>`

  Table 5-3 provides a detailed description of the arguments to ovcfgtopodata.

Table 5-3 Adding Upper Topology Instances

Parameter	Description
Parent FDN	The FDN of the parent object instance that contains this managed object instance.
Managed Object Class Name	This is the name of the managed object class type associated with this managed object instance.
RDN	The relative distinguished name of the managed object instance.
Shortname	An assigned alpha identifier for the managed object instance. This name is does not have to be unique.
Domain	The domain name for this instance and all its descendants.

Table 5-3 Adding Upper Topology Instances

Parameter	Description
XML File	The name of the configuration file to which topology instances are to be added.

- To add a connection instance definition to TopoData.xml, execute:

  `ovcfgtopodata -addCX <FDN> <MOC name> <RDN> <shortname> <from_FDN> <to_FDN> [domain] <xml_file>`

  The information to be entered is detailed in Table 5-4:

Table 5-4 Adding Connection Instances

Parameter	Description
FDN	The FDN of the parent object instance that contains this managed object instance.
Managed Object Class Name	This is the name of the managed object class type associated with this managed object instance.
RDN	The relative distinguished name of the managed object instance.
Shortname	An assigned alpha identifier for the managed object instance. This name is does not have to be unique.
From FDN	The fully distinguished name of the termination point instance from which the connection originates.
To FDN	The fully distinguished name of the termination point instance at which the connection to the second object is made.
Domain	The domain name for this instance and all its descendants.
XML File	The name of the configuration file to which topology instances are to be added.

Removing Managed Object Instances

To remove a network managed object instance from TopoData.xml, execute:

```
ovcfgtopodata -remInst <FDN> <xml_file>
```

Table 5-5 provides a detailed description of the arguments to ovcfgtopodata.

Table 5-5 Removing Upper Topology Instances

Parameter	Description
FDN	The FDN of the parent object instance that contains this managed object instance.
XML File	The name of the configuration file to which topology instances are to be added.

Updating Topology Instance Data

After managed object instance definitions are added or removed, TopoData.xml must be validated, deployed, and applied to the topology server before the new configuration data is functional.

- After entering the network object model data in TopoData.xml, validate and deploy the XML file. The validation process checks to see if the content you entered is consistent and valid against `TopoModel.dtd`. The deployment process generates target configuration files and stores them in `generated/` under the project directory.

  Execute: `ovcfgdeploy -topodata <project>`

  This command first checks the consistency of managed object instance data. The validation checks to see whether:

- To pull the upper topology configuration data stored on the OVO server to the topology server for processing by OV Topology Server, use the `ovtopodata.apply` command.

  On the topology server, execute:

  ```
  ovtopodata.apply <project>
  ```

Managing Agent Configuration

All agent instance definitions are stored in multiple Agent.xml configuration files. One Agent.xml file is defined per agent system connected to the OVO management server. These XML files contain the definitions to the lower topology instances needed to configure the lower topology of a telecom network as well as the definitions for the data collectors and sources belonging to a managed network.

This chapter describes the utilities and procedures to:

- Add data collector definitions to the topology.
- Remove data collector definitions.
- Add lower topology instances in the topology server database.
- Remove lower topology instances in the topology server database.
- Update agent configuration.

Adding Data Collectors

An *Agent*.xml consists of at least six tables you must complete:

- DCs
- Sources
- Source Details (One table for each source detail defined in the Sources table.)
- Equipment List
- Record Formats
- Lookup Tables (One for each lookup table definition needed.) (optional)

Adding Data Collectors

Complete the definition of the data collectors. This data forms the base of the rest of the data collection information to be configured.

To add a data collector definition to *Agent*.xml, execute:

```
ovcfgagent -addDC <DC name> <source> [-<dcdetails>] xml_file
```

Table 5-6 provides a detailed description of the parameters of ovcfgagent.

Table 5-6 Adding DCs

Parameter	Description
DC Name	The unique name for the data collector inside the Agent. Enter an alphanumeric name (up to 32 characters) with initial alpha character. Hyphens and underscores are allowed.
Source	A list of the input source names for the data collector.
DC details	Optional. Provides a list of data collector details associated with the data collector.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

IMPORTANT

Data collectors need to be registed with OVO in order to function properly. By default, one data collector per agent is registered for you. When you enter more than one definition of a data collector in an agent configuration file, OVO outputs a message asking which data collector should be started. For more information on configuring more than one data collector on an agent, see the white paper on configuring multiple data collectors..

Adding Sources

Input sources are defined in *Agent*.xml by five required attributes:

- Name
- Source Type
- Source Detail
- Status
- Equipment

To add an input source definition to *Agent*.xml, execute:

```
ovcfgagent -addSource <name> <source type> <source details>
<status> <equipment> xml_file
```

Table 5-7 provides a detailed description of the parameters of ovcfgagent.

Table 5-7 Adding Sources

Parameter	Description
Name	The name for the input source. Enter an alphanumeric name (up to 32 characters) with initial alpha character. Hyphens and underscores are allowed.
Source Type	The type of source; enter either `TCP` or `FIFO`.
Source Details	A list of source detail names associated with this input source.
Status	Indicates whether this input source is enabled to receive alarms or not; enter `Yes` or `No`.
Equipment	A list of equipment types that can be expected to provide data to this source.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

Adding Source Details

Source details define connections details, consisting of possible name-value pairs for the source. If a source detail is not already defined, then a new one is created. Otherwise, name-value pairs are appended to the existing table.

Possible entries for *`Name`* are:

- `FIFOName` – Name of the FIFO source to read. It only applies to FIFO sources.
- `TCPMode` – Mode in which the data collector is to operate. The data collector can act as either a `Server` or a `Client`. When the data collector is identified as a `Client`, it connects to the port on the remote host specified by `TCPHost`. When the data collector is identified as a `Server`, the host connects to its port specified in `TCPPort`.
- `TCPHost` – Name of the remote host to be connected to the data collector, when TCP functionality is in client mode.
- `TCPPort` – The IP port that is to be used for the TCP connection.

To add a source detail definition to *Agent*.xml, execute:

`ovcfgagent -addSourceDetail <source detail name> <name> <value> xml_file`

Table 5-8 provides a detailed description of the parameters of ovcfgagent.

Table 5-8 Adding Source Details

Parameter	Description
Source Detail Name	Name of input source detail.
Name	The name of the detail entry.
Value	The value of the detail entry.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

Adding Equipment List

To add an equipment definition to *Agent*.xml, execute:

`ovcfgagent -addEquip <name> <status> <record format> xml_file`

Table 5-9 provides a detailed description of the parameters of ovcfgagent.

Table 5-9 Adding Equipment List

Parameter	Description
Name	The name of the equipment type.
Status	Indicates whether the equipment is enabled to receive alarms; enter `Yes` or `No`.
Record Format	A list of record formats which describe the initial filters (begin/end) strings used for this piece of equipment. Note that more than one record format can be identified for each equipment type.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

Adding Record Formats

This section describes the information required for recognizing and classifying the messages received from an input source (the network element class that emitted the message).

To add a record format definition to *Agent*.xml, execute:

`ovcfgagent -addRecordFormat <name> <begin> <end> xml_file`

Table 5-10 provides a detailed description of the parameters of ovcfgagent.

Table 5-10 Adding Record Formats

Parameter	Description
Name	The name of the record format.
Begin	The beginning filter string for this record format. This is the regular expression that identifies the beginning of the message packet. This pattern can be alphanumeric with a maximum of 128 characters. This information is not mandatory. However, one of the two columns--Message Header Pattern or the Message Trailer Pattern--must have information entered.
End	The end filter string for this record format. This is the regular expression that indicates the end of the message packet. This pattern can be alphanumeric with a maximum of 128 characters. This information is mandatory if no information is entered in the Message Header Pattern column.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

Adding Lookup Tables

To add a new lookup table definition to *Agent*.xml, execute:

`ovcfgagent -createLookupTable <name> <column name> <column name> ...`

Table 5-11 provides a detailed description of the parameters of

ovcfgagent.

Table 5-11 Adding Lookup Tables

Parameter	Description
Name	Unique identifier for the lookup table.
Column Name	Identify a name for a column in the lookup table

To add a new entry to a lookup table in *Agent*.xml, execute:

```
ovcfgagent -addLookupTable <name> <column value> ...
```

Removing Data Collector Definitions

To remove a data collector definition, you may or may not need to remove supplemental definitions associated with the data collector, including:

- Sources
- Source details
- Equipment list
- Record formats
- Lookup tables (optional)

Removing Data Collectors

To remove a data collector definition from *Agent*.xml, execute:

```
ovcfgagent -remDC <DC name> xml_file
```

Table 5-12 provides a detailed description of the parameters of ovcfgagent.

Table 5-12 Removing DCs

Parameter	Description
DC Name	The unique name for the data collector inside the Agent. The data collector must already exist.
XML File	The name of the agent XML configuration file for which the data collector definition is to be removed.

IMPORTANT Data collectors are registed with OVO. When a data collector is removed, you should also remove the registration with OVO. For more information, see the white paper on configuring multiple data collectors.

Removing Sources

To remove an input source definition from *Agent*.xml, execute:

`ovcfgagent -remSource <name> xml_file`

Table 5-13 provides a detailed description of the parameters of ovcfgagent.

Table 5-13 Removing Sources

Parameter	Description
Name	The name for the input source. The input source must already exist.
XML File	The name of the agent XML configuration file from which the data collector definition is to be removed.

Removing Source Details

To remove a source detail definition from *Agent*.xml, execute:

`ovcfgagent -remSourceDetail <source detail name> xml_file`

Table 5-14 provides a detailed description of the parameters of ovcfgagent.

Table 5-14 Removing Source Details

Parameter	Description
Source Detail Name	Name of input source detail. The name-vlaue pair definition under this source detail are also removed.
XML File	The name of the agent XML configuration file from which the data collector definition is to be removed.

Removing Equipment List

To remove an equipment definition from *Agent*.xml, execute:

`ovcfgagent -remEquip <name> xml_file`

Table 5-15 provides a detailed description of the parameters of `ovcfgagent`.

Table 5-15 Removing Equipment List

Parameter	Description
Name	The name of the equipment type. This does not remove the record formats associated with the equipment type.
XML File	The name of the agent XML configuration file from which the data collector definition is to be removed.

Removing Record Formats

To remove a record format definition from *Agent*.xml, execute:

`ovcfgagent -remRecordFormat <name> xml_file`

Table 5-16 provides a detailed description of the parameters of `ovcfgagent`.

Table 5-16 Removing Record Formats

Parameter	Description
Name	The name of the record format.
XML File	The name of the agent XML configuration file for which the data collector definition is to be added.

Removing Lookup Tables

To remove a lookup table definition from *Agent*.xml, execute:

`ovcfgagent -remLookupTable <name>`

Table 5-17 provides a detailed description of the parameters of

ovcfgagent.

Table 5-17 Removing Lookup Tables

Parameter	Description
Name	Unique identifier for the lookup table.

Adding Lower Topology Instances

- **Network Element Instances**

  To add a network element object instance to *Agent*.xml, execute:

  ovcfgagent -addNE <RDN> <MOC name> <shortname> <domain> [<*timezone*>] <xml_file>

  This command adds a new network element definition into the agent topology.

  Table 5-18 provides a detailed description of the arguments of ovcfgagent.

Table 5-18 Adding Network Element Instances

Parameter	Description
RDN	The relative distinguished name of the managed object instance.
Managed Object Class Name	This is the name of the managed object class type associated with this managed object instance.
Shortname	An assigned alpha identifier for the network element instance. This name has to be unique among all network elements across the managed network. Validation of uniqueness is performed within an agent, but not across agents.
Domain	You may add a new domain name for this network element. By default, the network element inherits all of the domains from its parent. If no new domain name is to be added, use “” in this column.
Time zone	The time zone in which the network element is physically located.

Table 5-18 Adding Network Element Instances

Parameter	Description
XML file	The name of the agent XML file where network elements instances are to be added.

NOTE

The shortname assigned to this network element is important to remember. In the Mappings.xml file, the shortname is mapped to the fully distinguished name (FDN) of the upper topology object for this network element to create the complete FDN for the network element.

- **Managed Object Instances**

  To add a managed object instance to *Agent*.xml, execute:

  ```
  ovcfgagent -addMOI <parent RDN> <RDN> <MOC> [<shortname>]
  <xml_file>
  ```

  Table 5-19 provides a detailed description of the arguments of ovcfgagent.

Table 5-19 Adding Managed Object Instances

Parameter	Description
Parent RDN	The RDN of the parent managed object instance that contains this component object instance. The RDN path starts with the RDN of the network element in this agent for this component and ends with the RDN of the parent component, if any.
RDN	The relative distinguished name for this component object instance. The RDNs must be unique among siblings.
Managed Object Class Name	This is the name of the managed object class type associated with this component object instance. The MOC type for this instance must either be a component or a terminal point.
Shortname	An assigned alpha identifier for the managed object instance. If a shortname is not entered then one is assigned the value of the naming attribute from the RDN.

Table 5-19 Adding Managed Object Instances

Parameter	Description
XML file	The name of the agent XML file where managed object instances are to be added.

- Connection Object Instances

  To add a connection instance definition to *Agent*.xml, execute:

  ```
  ovcfgagent -addCx <parent RDN> <RDN> <from> <to>
  [<shortname>]
  ```

  The information to be entered is detailed in Table 5-20:

Table 5-20 Adding Connection Instances

Parameter	Description
Parent RDN	The RDN of the parent object instance that contains this managed object instance. The RDN path starts with the RDN of the network element and ends with the RDN of the parent object instance.
RDN	The relative distinguished name of the managed object instance. The RDN must be unique among sibling instances.
Managed Object Class Name	This is the name of the connection managed object class associated with this managed object instance.
From	The relative distinguished name of the termination point instance from which the connection originates.
To	The relative distinguished name of the termination point instance at which the connection to the second object is made. The RDNs of the two connection points must be different.
Shortname	An assigned alpha identifier for the managed object instance. If a shortname is not entered then one is assigned the value of the naming attribute from the RDN.
XML file	The name of the agent XML file where managed object instances are to be added.

NOTE For those connections whose parents are network instances, use `ovcfgtopodata` to create the connection.

Removing Lower Topology Instances

To remove a network element object instance from *Agent*.xml, execute:

```
ovcfgagent -remMOI <RDN> ... <RDN> <xml_file>
```

This command removes network element definitions from the agent topology. This command is used to remove network element, termination point, component, and connection instances.

NOTE When a termination point instance is removed, the associated connection instances for that network element instance are not removed automatically.

Table 5-21 provides a detailed description of the arguments of ovcfgagent.

Table 5-21 Removing Managed Object Instances

Parameter	Description
RDN	The relative distinguished name of the managed object instance.
XML file	The name of the agent XML file from which network elements instances are to be removed.

Updating Agent Configuration

After entering the agent topology data in *Agent*.xml, validate and deploy the XML file. The validation process checks to see if the content you entered is consistent and valid against Agent.dtd. The deployment process generates target configuration files and stores them in `/generated/Agent` under the project directory.

Execute: `ovcfgdeploy -agent <project>`

To pull the agent topology configuration files stored on the OVO server to the respective server for processing, use the `ovagt.apply` and `ovtopodata.apply` commands.

On the topology server, execute:

```
ovtopodata.apply <project>
```

On the OVO server, execute:

```
ovagt.apply <project> -n <node1 node2 ...>
```

Managing OM Events

This section describes the use of the OM event generator.

OM Event Generator

The OM event generator is a command line utility that generates object management events from an ASCII file. This utility enables OM Event information to be imported to an event presenter window. The events can then be owned and the required action taken on them.

A prerequisite for this utility is the creation of an intermediate file from which the event information is updated in the OM event database. The OM event generator requires that this intermediate file has the extension `.omimport` and is available in the directory `$FMSVAR/import/om`.

Format of the Intermediate File

The OM event generator intermediate file (`*.omimport`) is the file used to load the OM event information into the database. It contains the details of the event being loaded. The file must be an ASCII text file. It can be created and edited using any standard text editor.

The import utility reads all lines between the begin of file (**`BEGIN_5.0`**) and end of file (**`END_5.0`**) indicators.

Between these two indicators, the event generator utility reads each line in the file as an event record. Each event record consists of several comma-separated fields. Some fields may contain multiple entries.

Table 5-22 lists and describes the fields of the OM event generator intermediate file.

NOTE

For each record, all field values must be delimited by a pair of double quotes. If the field has NULL value, it must be represented as a pair of double quotes (" ").

Commas are read as field separators; therefore, *this character must not be part of the field value.*

Table 5-22 Import File Field Description

Field	Description
MOC	32 Character - Alphanumeric Mandatory Indicates the OVC/Assurance managed object class of the object. The managed object class specified must be created using the FMS Configurator, if it has not already been created. If the object class does not exist, then event creation will fail.
FDNString	1024 Character - Alphanumeric Mandatory Indicates the fully distinguished name (FDN) of the OVC/Assurance object. The length of the FDN must not exceed 1024 characters.
EventTime	Date. Optional Indicates the event time. It is in the format `yyyy-mm-dd hh:mi:ss`. If the time is not specified, the current time on the FM Server is used for the event time.
EventType	Numeric. Valid value 0,1,2 Indicates the event type. The valid entries and their description are: 0 event_creation 1 event_deletion 2 attribute_value_change
SourceInd	Numeric. Valid values 0, 1, 2 Indicates the source of the event. The values represent: 0 resource_operation 1 management_operation 2 unknown

Table 5-22 Import File Field Description

Field	Description
AttributeValueChangelist	Alphanumeric. Optional field Is used to change the value of any of the object attributes. The syntax for entry in this field is as follows: /*label1*="[*old_value*1]""*new_value*2"[[/*label2*="*old_value*2""*new_value2*"]...] Where: *label* is the name of the field whose attribute or value is being changed. *old_value* is the current value of the field. *new_value* is the new value of the field. If an entry is made in this field, *label* and *new_value* are mandatory. The double quotes around the fields *old_value* and *new_value* are mandatory. Multiple attribute changes can be specified in the same format within one record.
Notification Id	Long integer. Optional field Specifies the notification ID of the event.
AdditionalText	1024 Character - Alphanumeric Provides additional information on the event.
AdditionalInfo	1024 Character - Alphanumeric Provides additional information on the event.

Record format

The syntax for the records in the intermediate file is:

- Object Type: X730OMEventData;
  - — Record Syntax:

    ```
    "MOC'", "FDNString", "EventTime", "EventType",
    "SourceInd", "AttributeValueChangeList",
    "NotificationId", "AdditionalText", "AdditionalInfo"
    <newline>
    ```

  - — Sample record for a node create function where some of the optional fields are NULL.

    ```
    "MSCMOC","/net-id=GSMnet/ne-id=MSC_A","1999-11-15
    13:45:20","2","1","/standbyStatus=""1"," ",
    "This is a attribute changelist event.","This event
    source indicator is management."<newline>
    ```

Mandatory Fields for Event Creation

The following are the mandatory fields for event generation records:

- MOC
- FDNString
- EventType

Generating OM Events

The OM event generator must be run as user `oemfadm` or `root` on the FM Server machine. To execute the utility, enter:

`fmsomgen -f` *`filename`*

The utility reads each OM event record from the specified file and generates the events. These events are processed by the FM Server and stored in the OM event database. The events can be viewed via the OM Event Presenter on the topology GUI.

Files Created During Event Import

The OM event generator creates two files while processing the event records from the import file in the `$FMSVAR/import/om` directory. *These files have the same name as the import filename but with different*

extensions. The files created are:

- **Successful Records File.** This file contains the list of event records that had been successfully imported. The extension for this file is `.processed_om`.
- **Failed Records File.** This file contains the list of records, if any, that could not be processed. The extension for this file is `.failop_om`.

  To process these records, you must:

  1. Correct the records.
  2. Rename this file with the extension `.omimport`.
  3. Run the OM event generator again with the corrected file.

Viewing the Trace and Log Information

The OM event generator writes informational messages regarding event generation. The trace level depends on the setting of the environment variable `$OVTRACE_LEVELS`. To set the trace level and retrieve trace information use the utilities `ovdumplog` and `ovlognotifs`. Filter on `OME` for messages relating to the OM event generator.

Managing Topology Maps

This chapter describes the utilities and procedures used to:

- Manage map domains using the map presenter.
- Create link object maps.

Using the Map Presenter to Manage Domains

Domains are typically created and deleted using the topology import and export utilities and added to the installation using the Operation Profile Configurator. Additionally, the map presenter can be used to manually manage the map.

The map presenter provides two domain management menu commands for the `oemfadm` user:

- `Domain:Add to Domain`
- `Domain:Remove from Domain`

Adding Objects to a Domain

To add objects to a domain:

1. Log on as `oemfadm` on the GUI Client.
2. In the map presenter, open the domain map to which the object(s) must be added.
3. Make the map editable.
4. Add the required object(s) to the map.
5. Select the newly added object(s) on the map.
6. Click `Domain:Add to Domain`.
7. Save the map. The new objects are now part of the domain represented by the open map.

WARNING **Skipping either of steps 5 or 6 may result in a map that is out of sync with the domain.**

Deleting Objects from a Domain

To delete object(s) from a domain:

1. Log on as `oemfadm` on the GUI Client.
2. In the map presenter, open the domain map from which the object(s) must be deleted.
3. Make the map editable.
4. Select the object(s) to be deleted from the map.
5. Click `Domain:Remove from Domain`.

   Deleting an object deletes all the child objects contained under it.
6. Click `Edit:Delete` to delete the object from the map.
7. Save the map.

   The list of network objects in the `Admin Panel` no longer includes the objects that have been removed.

WARNING **Skipping any of steps 4, 5, or 6 may result in a map that is out of sync with the domain.**

Creating Link Object Submaps

The topology import utility creates object server maps and domain maps. It does not, however, create client submaps for link objects. The `oemflinkmap` utility creates these link object submaps.

`oemflinkmap` creates client submaps that display the link and the two termination points that the link connects. Double clicking on the link in the Map Presenter opens the link client submap. There can be one or more links between two termination points. If there is a single, non-bundle link between the two termination points, then one link object client submap opens containing the link. If there are multiple links that form a bundle, then one link object client submap opens for each link. These client submaps are attached to the link on the Server Map that they represent as the default submap. A link must be marked as part of a bundle to be in the bundle map.

Run `oemflinkmap` after the topology import utility completes

successfully. Each time this function is executed, it processes the entire topology. It creates the link client submaps for all newly created links and updates the information from any deletions or modifications to the topology that affect existing link submaps.

`oemflinkmap` must be run by user `oemfadm`. It should not be run at the same time as `fmsopcfg`, `fmsmhimport`, `guidbbkup`, or `guidbrestore`.

Creating a Link Object Client Submap

To create the link object client submap:

1. Close any `oemfadm` topology GUI sessions that are open.
2. Load the object information.
3. Log on as `oemfadm` on the presentation database machine.
4. Enter the command:

   `oemflinkmapimport`

   As the utility executes, it displays messages to standard output.
5. After the entire file has been executed, check the file indicated by the environment variable `$OEMFLMAPOUTFIL` for any errors in execution.

   Optionally, use `ovdumplog` to diagnose the `oemflinkmapimport` errors:

   `ovdumplog -f *LMAP*`

   If there are errors, correct the errors and repeat Step 4.
6. Restart operator's topology GUI sessions or use `File:Revert` to reload maps from the presentation database.

Notes on Using the oemflinkmap Function

- After the `oemflinkmapimport` utility is executed, the map presenter must be restarted or reverted (`File:Revert`) to reload the map information and make the new maps available.
- The `IltCoMo2MapObjClassMapping` table is used to map between the Managed Object Class (MOC) names and the Map Object Class types that the Map Presenter understands. All MOCs for NODE or LINK object types must be defined in the `mocsymfile` used by `guidbmocsym`. If a node or link MOC is not defined in this file and a link submap is to be built for this link, the creation of that map fails

and an error is logged.

- If a link is part of a bundle, it should have the Collapsed flag set in the intermediate file for the topology import utility. If the link is not marked as Collapsed (bundled) when the `oemflinkmap` function is executed, then it is not part of the bundle.
- Errors that affect the creation of a ClientMap for a particular link are reported to `$`*`GUIDBVAR`*`/share/log/oemflinkmap.errout`. The number of links that failed the map creation and the number that are created are kept. The information in the `oemflinkmap.errout` file contains the FDN of the link map creation that failed and other error information that details the problem. The information in `oemflinkmap.errout` can be used to help check the validity of NODE, LINK, and MOC information previously entered into the system. The information in `ovdumplog` may be required in some cases to get the linked error information if the failure was the result some other process failing a request. All of this error information exists in the `ovdumplog`; the `oemflinkmap.errout` file presents an easier way of accessing most of that information.

  Many errors that occur do not cause the `oemflinkmap` function to abort, but may be fatal to creating a particular submap.

  To debug problems with the node and link information that is provided to `oemflinkmap`, the `ovdumplog` command can be used. The errors for `oemflinkmap` begin with LMAP for link map. The errors may be chained to other errors if the error occurred in a different component.

  If you are unable to diagnose the error using the message generated during execution on your standard display or the error log file, `oemflinkmap.errout`, then you can run `ovdumplog`.
- The `oemflinkmapimport` utility may be re-executed as necessary. This utility must be re-executed if a link submap creation fails.

  Ensure that `fmsmhimport` has successfully processed all the records in the import file before running `oemflinkmapimport`. If the object data is not present in both the topology and presentation databases, then `oemflinkmapimport` cannot create the link submaps. Each time this utility is executed, the entire topology is processed. Selective link submap creation is not supported.
- If the `oemflinkmapimport` command returns `l_err = 0` on completion, then no fatal errors were encountered. However, the

`oemflinkmap.errout` file still needs to be checked to see if there were any failures for a particular map or to see how many link submaps were created.

If the `oemflinkmapimport` command returns `l_err` equal to a value other than 0, then some fatal error was encountered. In which case, check the messages that were output on the terminal, and correct the errors. You can use the error file defined by the environment variable `$OEMFLMAPOUTFIL` for additional information in checking the error.

If you require additional help to diagnose the error, then run `ovdumplog` and check for errors that contain the string `LMAP`.

6 Configuring Operator Environments

This chapter describes the procedures to:

- Provide operator access to the iNOC Console.
- Create operation profiles.
- Assign operation profiles to users.

About the iNOC Console

The iNOC Console is a collection of graphical user interfaces through which operators:

- View all processed messages via the message browser.
- View correlated topology messages via the problems presenter.
- View the topology of the network and symbolic representation of the network objects via the map presenter.
- Set up and monitor outage plans for managed object instances on outage via the outage plan presenter.

These GUIs are described in the online help installed with the iNOC Console.

Prerequisites

The prerequisites for setting up the iNOC Console are:

- A completed installation and configuration of the OVO server.
- A completed installation and configuration of the topology server.
- A completed installation and configuration of databases.
- A completed installation of the iNOC Console components.

Smart Navigations

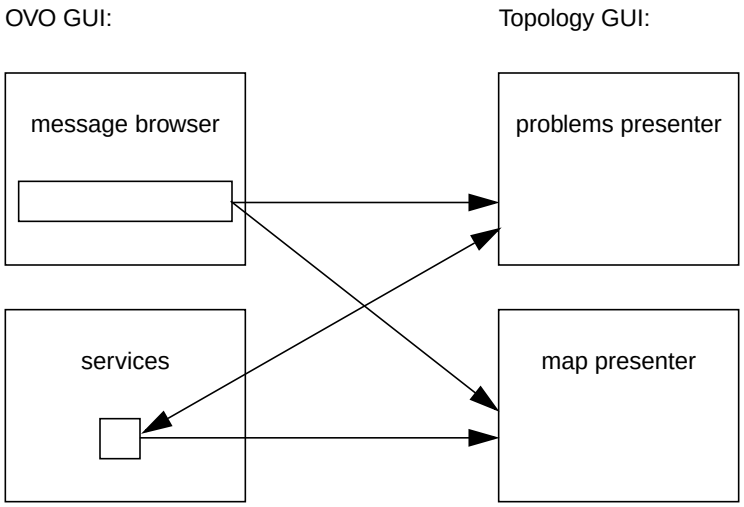
The following cross launches from the OVO operator GUI to the topology GUI are supported:

- Given a problem message in OVO, open a topology GUI map presenter containing the element or object that caused the problem.
- Given a problem message in OVO, show the events correlated in that problem and the problem history in the topology GUI problem presenter.
- Given a service in Service Navigator, open the topology GUI problem presenter with all the problems associated with elements mapped to that service.

- Given a service in Service Navigator, open a topology GUI map that shows the elements affecting that service.
- Given an element in a topology GUI map, show a list of services with which that element is associated.

Figure 6-1 shows the possible cross launches from the OVO operator GUI to the topology GUI.

Figure 6-1 Relationship Between GUIs



NOTE These smart navigations are only valid when the OVO operator GUI (Java GUI) is used. The OVO admin GUI (native GUI) does not permit launching to the topology GUI.

Users and Operation Profiles

The iNOC Console supports several types of users: integrators, administrators, and operators. Each user type may have one or more **user name**s defined to allow **users** to logon to the various GUIs: OVO operator GUI, OVO admin GUI, NNM admin GUI, or the topology GUI.

Users also have assigned to them **operation profiles**. These profiles

determine which applications, tasks, and objects users can access. Users can be assigned one or more operation profiles. In OVO, the concept of operation profiles is collective, meaning operators assigned to multiple operation profiles inherit all of the permissions of the collective profiles. In OV Topology Server, the concept of operation profiles is restrictive, menaing only one operation profile is accepted when launching the topology GUI.

Integrators and administrators can log onto the two iNOC Console using the default administrator user `opc_adm`. When the `opc_adm` user launches the topology GUI from the OVO operator GUI, he is, by default, logging onto the topology GUI as user `oemfadm` with operation profile `admin`. In OVSACN, this default administrator has full permissions and responsibilities.

Operators can log onto the iNOC Console using the default operator user `telco_op`. This user is assigned all topology GUI launching and navigation permissions. When `telco_op` user launches the topology GUI, he is logged onto the this GUI as user `telco_op` with operation profile `Telco_Op`.

It is the role of integrators/administrators to define additional user names and operation profiles.

NOTE The iNOC users must be configured with the same user name on both the OVO management server and the topology server in order to successfully launch and perform navigations from the OVO operator GUI to the topology GUI.

Topology GUI Login

When launching the topology GUI from the OVO operator GUI, OVSACN needs the following information defined:

- Topology server hostname – By default, the value of the topology server hostname is read from the first line of `hostfile.dat`.
- User name – By default, the user name is assumed to be the same as the OVO user name, except when OVO user is `opc_adm`. In this case, the default user name is `oemfadm`.
- User password – By default, the password is ignored.

NOTE

The default iNOC Console configuration relies solely on the OVO user and password authentication process to launch the topology GUI. If additional authentication is required to launch the topology GUI, then set the `FORCE_TELCO_GUI_LOGIN_AUTHENTICATION` environment variable to `Yes`. For more information about changing authentication, see “Adding Authentication” on page 178.

- Operation profile – By default, the operation profile is the first profiles in the user's assigned operation profile list.

About Configuring Operator Access

This section describes the tool used to configure new users and operation profiles on the topology server, the configuration files where additional users and operation profiles can be specified, and the iNOC default operator user and profile.

First, let's discuss the general process to configure operator access. To configure operator access to the iNOC Console:

- On the OVO management server system:
  - Create a new OVO user with the `User Profile Bank` window.
  - Create a new OVO user profile with the `User Bank` window.
  - Assign OVO user profiles to OVO user with the `User Bank` and `User Profile Bank`.
- On the topology server system:
  - Create OS user matching the name of the OVO user via SAM.
  - Create a new topology operation profile with the Operation Profile Configurator.
  - Create a new topology user matching the name of the OVO user with the Operation Profile Configurator.
  - Assign topology operation profiles to the topology user with the Operation Profile Configurator.

Operation Profile Configurator

Use the Operation Profile Configurator to define operation profiles to allow operator access to the data managed by the topology server. Operation profiles present specific information about the managed network via the topology GUI presenters.

The Operation Profile Configurator can be invoked and used only by the user `oemfadm` or any user with the same user access. Before running this utility, close any topology GUI presenters that are open under the user `oemfadm`.

To invoke the Operation Profile Configuration utility, execute:

fmsopcfg

This command opens the Operation Profile Configurator window shown in Figure 6-2. This window includes the window title and the menu bar. The text area of the window contains the name of the configuration utility and does not accept any text entries.

Figure 6-2 Operation Profile Main Menu Screen

The Operation Profile Configurator windows consist primarily of text boxes, lists, and the following buttons as applicable:

[Add]	Adds the name entered in the text box to the list.
[Delete]	Deletes the item selected from the list.
[Modify]	Modifies the properties of the item selected in the list.
[Apply]	Saves the unsaved changes made in the current session.
[Reset]	Reverts the configuration to the status of the last saved configuration. All changes made since the last Apply function are removed.
[Close]	Closes the current window.
[-->]	Moves the selected item from the left pane to the right pane.
[<--]	Moves the selected item from the right pane to the left pane.

The **Operation Profile Configurator** maintains user profiles that define operator access to the topology GUI presenters and a subset of the managed network and its associated messages. Use this configuration utility to configure:

- Operation Profiles

Filters, applications, tasks, management domains, and work schedules are assigned to operation profiles. Each user can have assigned multiple operation profiles.

- Filters

  Filters restrict the operator's access to problems based on the alarm type, probable cause, specific problem, and severity. Each user can be associated with one or more filters. Multiple filters can be associated with each operation profile.

- Applications and task sets

  These are a set of tasks to which the user can be assigned access. This defines all the applications and the tasks within the applications that are available for each operation profile. The application and tasks correspond to GUI application and tasks. Applications and their tasks can be defined into `application domains`. Multiple application domains can be linked to each operation profile.

- Management Domains

  The monitored topology is divided into logical management domains for operator assignments. The managed object (MO) management domain is a domain explicitly defined by the network administrator. Each MO management domain is associated with one or more managed objects. Multiple management domains can be associated with an operation profile.

Default Operator and Operation Profile

Telco_Op is the name of the preconfigured, default topology GUI operator operation profile. This profile provides its members access to topology problems, managed objects, and outages as well as the applications and tasks that can be applied to these objects.

Operators logged on as user `telco_op` with the profile `Telco_Op` see only topology-specific messages in the OVO message browser. From the OVO operator GUI, they can launch the topology GUI presenters and send demo alarms. From the topology GUI, operators have read-only access to the maps and all topology objects in the map presenter and nearly-full access to problems in the problems presenter. Operators may also view topology outage plans applied to network elements.

When OV Topology Server is optionally installed and configured, the following OVO configuration is evident:

- `Telco iNOC` application group is created that contains applications to launch the topology GUI and its presenters.
- `Telco_Op` profile is created and added to the User Profile Bank.
- `Telco iNOC` application group is assigned to the `Telco_Op` profile.
- `telco_op` user is created.
- `Telco_Op` profile is assigned to the `telco_op` user.
- `Telco_Op` profile is assigned to the `opc_adm` user.

When OV Topology Server is optionally installed and configured, the following topology server configuration is evident:

- `Telco_Op` profile is created that provides access to problems and outages associated with all managed objects in the domain.
- `telco_op` OS user is created.
- `telco_op` user name is created.
- `Telco_Op` profile is assigned to `telco_op` user.
- `Telco_Op` profile is assigned to `oemfadm` user.

NOTE The creation and assignment of the users and operation profiles must occur on each management server system in order for the iNOC Console to function as documented.

Configuration Files for Users and Profiles

The integrated login functionality of the iNOC Console simplifies and hides the specifics of the topology GUI login from users. However, two configuration files are provided with OV Topology Server to facilitate customizing the login process and changing the operation profile to be used.

topologin.user is an ASCII file that specifies the topology GUI login information. user is the name of the user logged onto the iNOC Console. This file is optional and must reside in the $HOME directory on Unix systems and in the %TEMP% directory on NT systems.

`topologin.`*`user`* is read before the topology GUI is launched and must contain all login information required by the topology GUI, including:

- Hostname of the topology server
- User name
- User password
- Operation profile for the topology GUI

topoprofile.user is an ASCII file that specifies the operation profile to be used when launching the topology GUI. user is the name of the user logged onto the iNOC Console. This file is optional and must reside in the $HOME directory on Unix systems and in the %TEMP% directory on NT systems.

The `topoprofile.`*`user`* file should contain a single line, specifying the name of the operation profile to be used. When a `topoprofile.`*`user`* file is not present, then the user is logged on with the first operation profile configured for that user.

NOTE If both the `topoprofile.`*`user`* and `topologin.`*`user`* files are present, then the `topologin.`*`user`* file takes precedence.

Adding iNOC Users

For access to the iNOC Console, users must set up the following information:

- A user name and password for access to the OVO operator GUI.
- A user name for access to the topology GUI that matches the OVO user name.
- An OS user on the topology server that matches the OVO user name.

Adding OVO Users

To add a new OVO user:

1. From the OVO management server, log on to the OVO admin GUI as user `opc_adm`.
2. Open the `User Profile Bank` window by clicking **`Window:User Profile Bank`**.
3. Optionally, define a new OVO user profile by:
   - Clicking **`Actions:User Profile -> Add`**
   - Assigning necessary message, application, and node groups to the new profile.
   - Assigning the `Telco iNOC` application group to the new profile, which allows the launching of the topology GUI presenters.
   - Saving the new profile
4. Click **`Window:User Bank`**.
5. Click **`Actions:User -> Add`**.
6. Assign a user name and password.
7. Assign appropriate capabilities to the new user.
8. Click [**`Profiles`**]. The `Profiles of New User` window displays.
9. Drag and drop user profiles from the `User Profile Bank` window to the `Profiles of New User` window.
10. Save and close all windows.

Adding a Topology GUI User

To access the topology GUI, a user must have the following information defined in the Operation Profile Configurator:

- OV User Name—The user ID for identifying the user within the topology server.
- User Name—The user's login ID. Each login ID is associated with a hostname.
- Host Name—The host name of the machine from which the user can log onto the topology GUI.
- Operation Profile—An operation profile must already exist in order to assign it to a user.

To create a new topology GUI user on the topology server:

1. From the topology server management server, log on as `oemfadm`.
2. Launch the Operation Profile Configurator, **`fmsopcfg`**.
3. Click **`User:User Maintenance`**.
4. Click [**`Add`**]. The `User Modification` dialog displays.
5. Click the `Operation Profile` tab to display the `User Association` page. Use this page to assign topology profiles to the new user.
6. Select an appropriate profile, and click **`-->`**.
7. Select the `Attributes` tab to display the `User Attributes` page.
8. Enter the name of the user in the `OVuser Name` text entry box.
9. Enter the topology server hostname for that user in the `Host Name` text entry box.
10. Click [**`Add`**]. This action adds the login details to the scroll box under the columns `User Name` and `Host Name`.
11. Click [**`Apply`**].
12. Close all windows and the Operation Profile Configurator.

Adding an OS User

An OS user name should be defined on the topology server with the same

name as the OVO user created on the OVO management server. To create a new OS user:

- Invoke SAM as user root.
- Double-click **Accounts for Users and Groups**.
- Double-click **users**.
- Click **Actions: Add**.
  - Enter the name of the OVO user in the Login Name field.
  - Enter the next available value for the User ID field.
  - Enter an appropriate directory for the Home Directory field.
  - Enter **oemf** in the Primary Group Name field.
  - Enter an appropriate value for the Start-Up Program field.
  - Enter an appropriate password in the Password field.
- Exit **Users**.
- Double-click **Groups**.
- Select Users group name.
- Click **Actions: Add**.
  - Enter the OS user name in the Login Name text entry box.
  - Click [**Add**].
  - Repeat as needed.

NOTE When the iNOC Console is running on Unix systems, the OS user name must also be a member of the oemf OS group in order to launch the topology GUI.

Deleting a User

To delete an operator's access to the iNOC Console:

- On the OVO management server, delete the user name from the User Bank window.

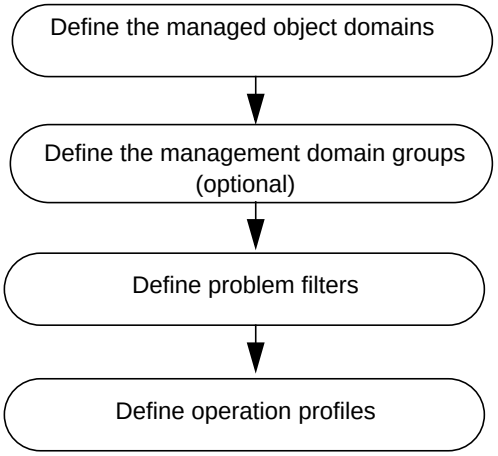
- On the topology server, delete the user name from the list of users in the Operation Profile Configurator using the function `User:User Maintenance`.
- On the topology server, remove the operator's HP-UX login ID from the topology server host.

Configuring Operator Access

Operators use the iNOC Console to monitor the function of the network and trigger applications based on the state of the network. Each operator may monitor and manage a different part of the network depending on how administrators distribute the workload of the managed network. This distribution is accomplished through **operation profiles**, or roles. Operation profiles determine which applications, tasks, and objects operators are permitted to access.

Figure 6-3 presents a flowchart of the planning steps for configuring topology access that must be completed before entering the configuration into the system. See the OVO manuals for configuring operator access to the OVO operator GUI.

Figure 6-3 Steps in Planning Operator Access to Topology GUI



Managed object domains. Managed object domains make it possible to assign multiple managed objects to an operation profile in one action. A managed object domain consists of a set of managed objects grouped together for management purposes. Any number of objects can belong to the same managed object domain. All child objects of these managed objects also belong to the same managed object domain.

Management domain groups. Like managed object domains, management domain groups make it possible to assign multiple

managed objects to an operation profile in one action. The management domain group is a management group of managed object domains. It is created only for ease of management in assigning domains to operation profiles.

Problem filters. Problem filters define the nature of alarms that operators can view.

Operation profiles. Managed object domains, management domain groups, applications, tasks, and filters are grouped together to form an *operation profile*.

Define the Managed Object Domains

A managed object domain consists of a set of managed object instances grouped together for management purposes.

You can define managed object domains using the `Domain` parameter in the `TopoData.xml` configuration file. For information on defining managed object domains with `TopoData.xml`, see *HP OpenView Service Assurance for Communication Networks Configuration Guide.*

Alternatively, you can define managed object domains with the Operation Profile Configurator. All domains defined using the `TopoData.xml` file appear in the Operation Profile Configurator in the `Available MO Management Domains` list box.

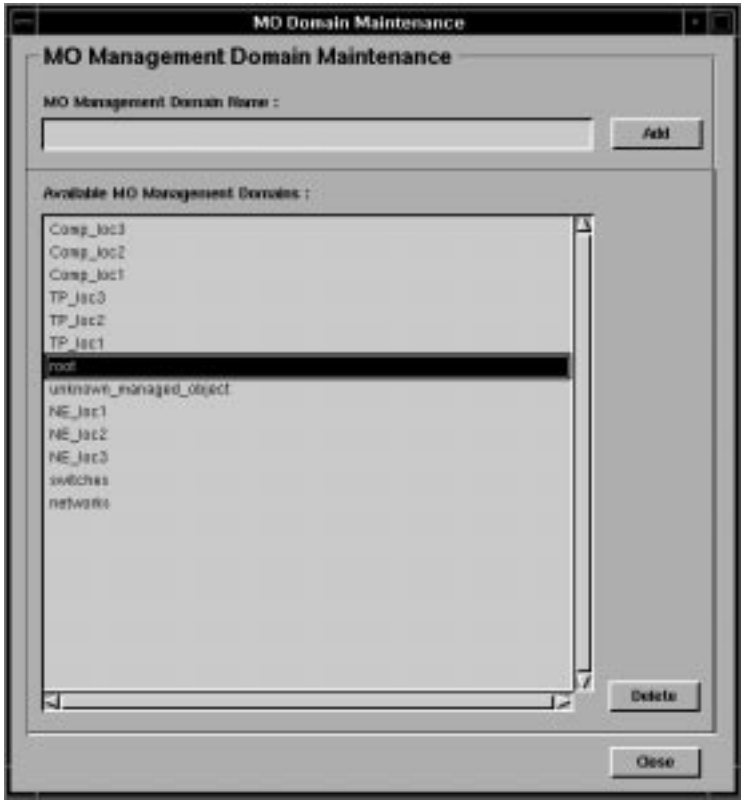
When using the Operation Profile Configurator, the information listed in Table 6-1 is required.

Table 6-1 Managed Object Domains

Parameter	Description
Managed Object Domain Names	This is the name of the managed object domain. It is suggested that logical names, by location or object type, be used for easy identification of the domain. *Note that you must not assign MOC names or managed object instance names as managed object domain names.* There is no restriction on the number of domains that can be created.
Managed Object Domain Object Names	These are the names of the managed objects contained within the managed object domain. These are the object instances as seen on the map presenter. You can list them by their shortname or map label. There is no restriction on the number of managed objects that can belong to the same domain. However, a managed object must be assigned to only one managed object domain. Any number of objects can belong to the same managed object domain. All child objects of these managed objects also belong to the same managed object domain.

To add a managed object domain, click `ManagementDomain:ManagedObject`. The window shown in Figure 6-4 displays.

Figure 6-4 Managed Object Management Domain Maintenance Window



You can add and delete management object domain names from the `Managed Object Management Domain Maintenance` window. Any action executed in this screen is immediately saved.

Two default domains, `unknown_managed_object` and root, are created during the installation process. The `unknown_managed_object` domain contains any managed object specified in a problem message that is not present in the topology database. The `root` and `unknown_managed_object` managed object domains are assigned to the `admin` operation profile as well as the `Telco_Op` operation profile.

All domains defined using the `TopoData.xml` file also appear in the `Available MO Management Domains` list box.

Adding a Managed Object Domain

To add a managed object domain:

1. Enter the name of the management domain in the `Managed Object Management Domain Name` text entry box in the `Managed Object Management Domain Maintenance` window.
2. Click **`[Add]`**.

   This action creates the management domain and adds its name to the `Available MO Management Domains` list box.

When a management domain is created in the Operation Profile Configurator, a client map named `domain-name_domain` is created in the map presenter. This map is also referred to as Managed Object Management Domain Map. It is a shared client map and is maintained by the user `oemfadm`.

Deleting a Managed Object Domain

To delete a managed object domain:

1. Select the name of the management domain to be deleted from the `Available MO Management Domains` list in the Managed Object Management Domain Maintenance window.
2. Click **`[Delete]`**.

   The selected management domain name is deleted from the list of management domains.

When a management domain is deleted in the Operation Profile Configurator, its domain map is deleted from the map presenter.

Define Management Domain Groups (optional)

A management domain group is a logical grouping of managed object domains. This grouping is specifically used to assign managed object domains to operation profiles and has no other relevance. *Creating and using management domain groups is optional.*

When defining management domain groups, the information listed in Table 6-2 is required.

Table 6-2 Management Domain Group

Information Head	Description
Management Domain Group	List a set of unique names for the management domain groups.
Managed Object Domain Names	These are the names of the managed object domains that belong to the management domain group defined above. These names should have been listed in the Managed Object Domains Worksheet completed in Step 1. A managed object domain can belong to more than one management domain group. There is no limit on the number of managed object domains that can belong to a management domain group.

To add a management domain group, click `ManagementDomain:Group`. The window shown in Figure 6-5 displays.

Figure 6-5 Management Domain Group Maintenance Window

Use the `Domain Group Maintenance` window to add, delete, and modify management domain groups. You can also copy an existing management domain group and create another group by making modifications to it. Any action executed in this screen is immediately saved.

Adding a Management Domain Group

Adding a management domain group is a two step process. Create the management domain group by adding its name. Then associate the required managed object management domains to the management domain group.

To create a management domain group:

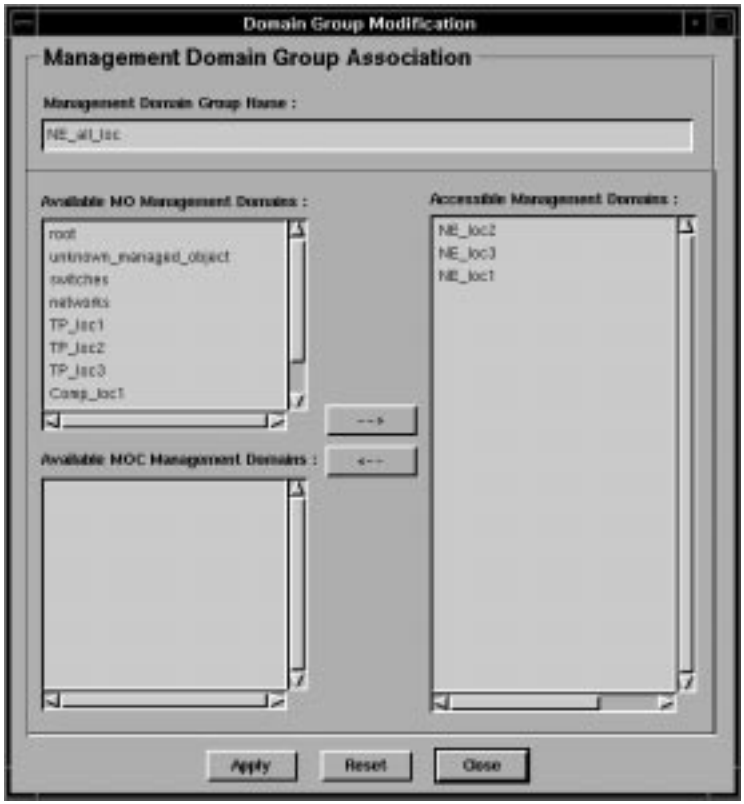
1. Enter the name of the management domain group in the `Management`

`Domain Group Name` edit box.

2. Click **[Add]**.

   This action creates the management domain group name in the text box and opens the `Management Domain Group Association window` shown in Figure 6-6.

Figure 6-6 Management Domain Group Association Window



This window displays the list of managed object management domains that are available.

To associate the managed object management domains with the management domain group:

1. Select the managed object management domains by name from the `Available MO Management Domains` list.

Multiple managed object management domains can be selected by pressing **Shift** and clicking on the required domain names or by clicking within the list and dragging the mouse through the list of required names.

2. Click **[-->]** to associate the managed object management domains with the specified management domain group.

   This action adds the managed object management domain name to the Accessible Management Domains list.

3. Click [**Apply**] at the bottom of the window to save the association of the managed object management domains with the specified management domain group.

To disassociate any of the associated domains from the management domain group:

1. Select the managed object management domains by name from the Accessible Management Domains list.

   Multiple managed object management domains can be selected by pressing **Shift** while selecting the domains or selecting a domain name and dragging the mouse through the list.

2. Click **[<--]**.

   This action removes the selected management domain names from the Accessible Management Domains list and returns them to the Available MO Management Domains list.

3. Click **[Apply]** at the bottom of the window to save the deletions made in the list of management domains associated with the specified management domain group.

To undo a change made to the list of management domains associated with the specified management domain group:

1. Click **[Reset]**.

   This action un-does all changes (additions or deletions) made to the management domains associations since the last time [Apply] was selected.

To close this window and return to the Management Domain Group Maintenance window, click **[Close]**.

You are prompted to confirm exit if you have not *applied* all of your changes.

Modifying a Management Domain Group

You can modify the management domains associated with a management domain group at any time. To modify a management domain group:

1. Select a management domain group name from the `Available Management Domain Groups` list in the Management Domain Group Maintenance window.
2. Click **`[Modify]`**.

   This action opens the `Management Domain Group Association window` with the `Accessible Management Domains` list showing the currently associated management domains.

   The selected management domain group name appears in the `Management Domain Group Name` text box. This name cannot be edited. The `Accessible Management Domains` list shows the managed object management domains associated with the selected management domain group.
3. Make the required changes to the associated management domains list as you did while adding the domains.
4. Click **`[Apply]`** to save the modified associations with the selected management domain group.

Copying a Management Domain Group

You can copy an existing management domain group to a new management domain group name. This could be useful when creating management domain groups that are similar or have overlapping managed object management domains.

To copy a management domain group:

1. Select the name of the management domain group to be copied from the `Available Management Domain Groups` list in the Management Domain Group Maintenance window.
2. Click **`[Copy]`**.

   A dialog box prompts you to name the new management domain group.
3. Enter a unique name for the new management domain group.
4. Click **`[Apply]`** to create the new management domain group with the

same domain association as the original one.

The new name appears in the Available Management Domain Groups list. You can now select the new management domain group and modify its management domain.

Clicking [Cancel] closes the dialog box without adding a new management domain group.

Deleting a Management Domain Group

To delete a management domain group:

1. Select the name of the management domain group to be deleted from the Available Management Domain Groups list in the Management Domain Group Maintenance window.
2. Click **[Delete]**.

   The name of the selected management domain group is deleted from the list of management domains.

Deleting a management domain group from the Operation Profile Configurator does not affect the managed object management domains associated with that management domain group.

Define Problem Filters

Within OV Topology Server, access to network problems can be defined using **problem filters**. Problem filters are defined by the following:

- **Alarm Type**. You can assign access to any or all of the five X.733 defined event types. If event types are selected as attributes, the operation profile provides access to only those alarms with the specified event type.
- **Probable Cause Type**. OV Topology Server provides message mapping by global and local variables. If probable cause types are selected as attributes, the operation profile provides access to only those alarms with the specified probable cause.
- **Probable Cause**. Among the alarm types assigned, probable causes can be selectively accessed.
- **Specific Problems**. If specific problems are selected as attributes, the operation profile provides access to only those alarms with the specified specific problem. These specific problems are defined in the Topo-Smart templates.
- **Alarm Severity**. Operation profiles can be associated with specific severity levels. If severity levels are selected as attributes, the operation profile provides access to only those alarms with the specified severity level.

Table 6-3 provides a description of the information needed to configure problem filters.

Table 6-3 Problem Filters

Information Head	Description
Filter Name	This is the name of the filter. It is internal to the Operation Profile Configurator. Enter any unique name to identify the filter you are defining.
Alarm Type	One of five X.733 event types. More than one event type can be assigned to a filter.

Table 6-3 Problem Filters

Information Head	Description
Probable Cause	This the probable cause that relates to the event type specified in the previous column. For each probable cause, indicate whether it is a global variable or a local one. Multiple probable causes can be selected.
Specific Problem	This is the specific problem that relates to the probable cause that is specified in the previous column. Multiple specific problems can be specified.
Severity	One of six X.733 severity levels. Multiple levels of severity can be associated with one operation profile.

To add a problem filter definition, click **`Filter:ProblemFilter Maintenance`** from the Operation Profile Configurator. The `Problem Filter Maintenance` window displays as shown in Figure 6-7.

Use the `Problem Filter Maintenance` window to add, delete, and modify problem filters. You can also copy an existing filter to create a new one by making modifications to it. Any action executed in this screen is immediately saved.

Figure 6-7 Problem Filter Maintenance Window

Adding a Problem Filter

Adding a problem filter is a two step process. Create the filter by adding its name. Then define the filter by the alarm type, probable cause, and severity details.

To create a problem filter:

1. Enter the name of the problem filter in the `Problem Filter Name` edit box.
2. Click **[Add]**.

   This action adds the problem filter name to the list and opens the `Problem Filter Association window`.

This screen has two tabs:

- `OperationProfile`

  Use this tab to associate the current problem filter with existing operation profiles. For information on using this tab, see “Associating Problem Filters with an Operation Profile” on page 173.

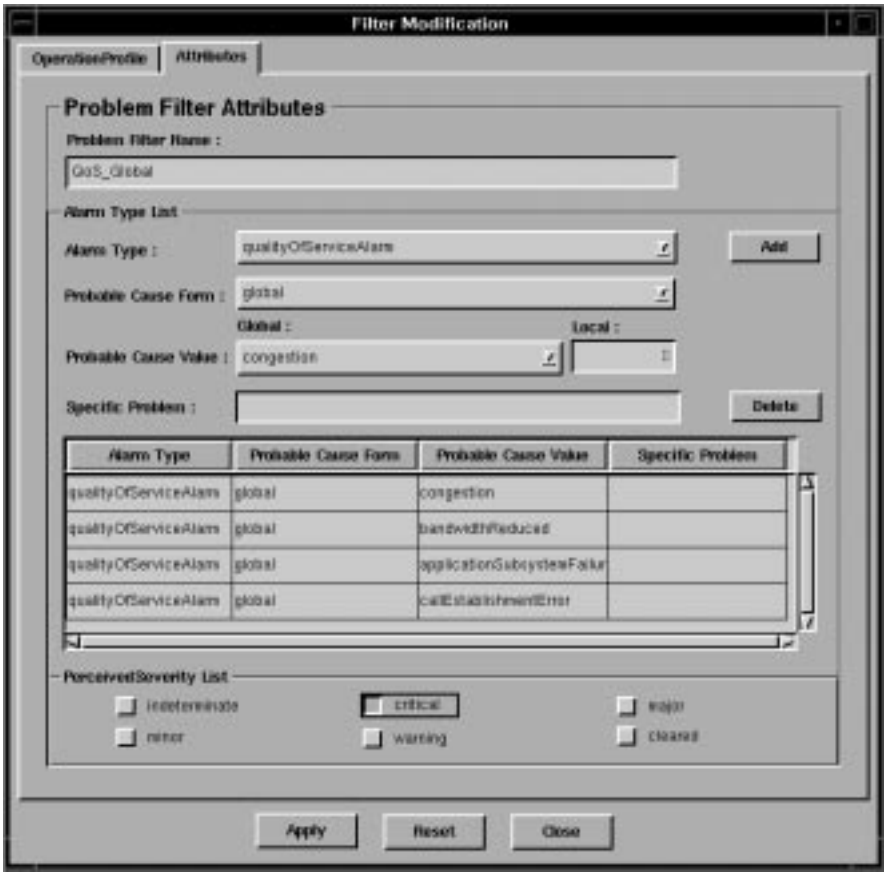
- `Attributes`

  Use this tab to define the problem filter. This tab is described below.

3. Click the `Attributes` tab to define the attributes for the problem filter.

   The window shown in Figure 6-8 displays. Use this window to define the alarm type, probable cause, and severity level associated with this problem filter.

Figure 6-8 Problem Filter Attributes Window



To associate the alarm type for this problem filter:

1. Select the alarm type from the `Alarm Type` list.

   The list contains the five X.733 event types.

2. Select the probable cause type from the `Probable Cause Form` list.

   The list contains two options: global and local. Select `global` to associate this problem with X.721 probable causes.

   Select `local` to associate this problem with M3100 probable causes.

3. Depending on the probable cause form you selected in step 2, select the probable cause value. Multiple selections are allowed.

   If you selected `global` in step 2, select the probable cause values from

the list under the `Global:` section of the `Probable Cause Value`.

If you selected `local` in step 2, enter the local probable cause values under the `Local:` section of the `Probable Cause Value`.

4. Click **[Add]** to save the association of the alarm type with the problem filter.

   The alarm type list details are displayed in the scroll box in table form.

   Repeat the above steps for all alarm type settings for this problem filter.

5. Use the `Perceived Severity List` to select the levels of alarm severity to add to the problem filter. This setting applies to all alarm types associated with the problem filter.

6. Click **[Apply]** in this window to save the additions made to the problem filter attributes.

To disassociate any of the associated alarm type list settings from the problem filter:

1. Select the alarm type name from the list. The line is displayed in black to indicate its selection.

2. Click **[Delete]**.

   This action removes the selected alarm type from the list.

3. Click **[Apply]** at the bottom of the window to save the deletions made in the alarm type list of problem filter attributes.

To undo a change made to the list of alarm types associated with the specified problem filter, click **[Reset]**.

This action un-does all changes (additions or deletions) made to problem filter attributes since the last time you clicked **[Apply]**.

To close this window and return to the `Problem Filter Maintenance` window, click **[Close]**.

You will be prompted to confirm exit if you had not *applied* any of changes you had made.

Modifying a Problem Filter

After you have created a problem filter, you can modify the tasks associated with it whenever required. To modify a problem filter:

1. Select the problem filter name from the Available Problem Filter Names list in the Problem Filter Maintenance window.
2. Click **[Modify]**.

   The Problem Filter Modification Window appears.
3. Click the Attributes tab to open the Alarm Type List setting.
4. Make the required changes to the problem filter attributes as you did while adding the attributes.
5. Click **[Apply]** to save the modified attributes of the selected problem filter.

Copying a Problem Filter

You can copy an existing problem filter to a new problem filter name. This could be useful when creating problem filters that are similar or have overlapping attributes.

To copy a problem filter:

1. Select the name of the problem filter to be copied from the Available Problem Filter Names list in the Problem Filter Maintenance window.
2. Click **[Copy]**.

   A dialog box prompts you to name the new problem filter.
3. Enter a unique name for the new problem filter.
4. Click **[Apply]** to create the new problem filter with the same attributes as the original one.

   The new name appears in the Available Problem Filter Names list. You can now select the new problem filter and modify its attributes.

   Clicking **[Cancel]** closes the dialog box without adding a new problem filter.

Deleting a Problem Filter

To delete a problem filter:

1. Select the problem filter to be deleted from the Available Problem Filter Names list in the Problem Filter Maintenance window.

2. Click **[Delete]**.

   The name of the selected problem filter is deleted from the list of problem filters.

Define Operation Profiles

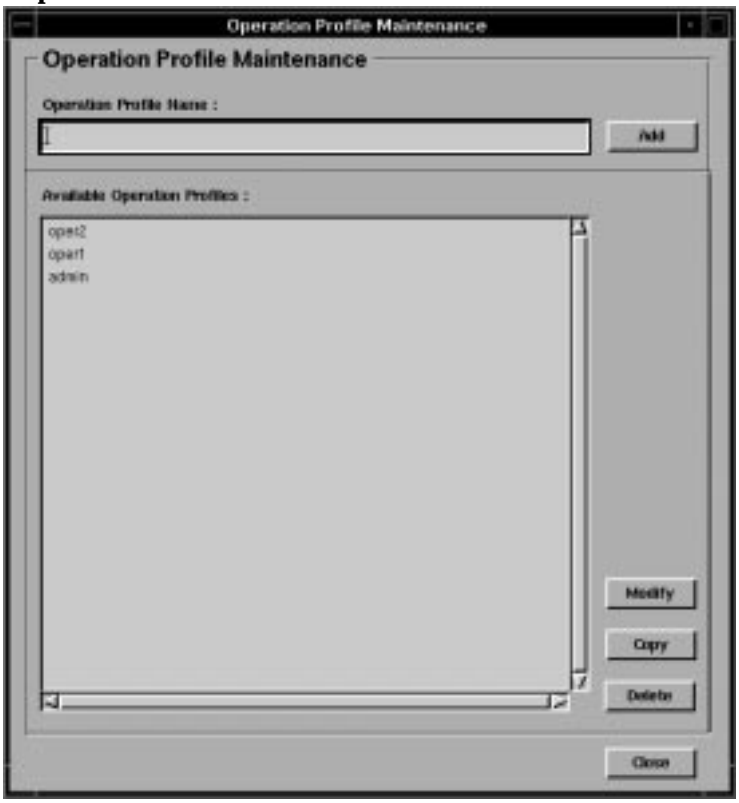
Managed object domains, management domain groups, applications, and filters are grouped together to form an *operation profile*. The types of information required to define an operation profile is described in Table 6-4.

Table 6-4 Operation Profiles

Information Head	Description
Operation Profile Name	This is a unique name for the set of functions for which access is being defined.
Application Name	A set of four applications, each providing various tasks. Operators need access to these tasks to be able to execute them. Access to the applications is assigned to the operators through the operation profile. Multiple applications can be linked to a single profile.
Task Name	These are the tasks that are associated with the application name specified in the previous column. Select the desired tasks for this operation profile. **NOTE** All operation profiles must include the `managed_object_application` with the task `get`.
Management Domain Group Name	This is the name of the management domain group. Multiple management domain groups can be linked to a single profile. If, for any management domain group, you do not want to include *all* the linked managed object domains, then you can list those you want in brackets. These can be individually assigned from the management domain group.
Managed Object Domain Name	Use this column to specify any managed object domain that you want to link to this profile that is not covered by the management domain group specified in the previous column.
Filter	Multiple filters can be linked to a single profile.

To add an operation profile definition, click **`OperationProfile:Operation Profile Maintenance`**. The `Operation Profile Maintenance Window` shown in Figure 6-9 displays.

Figure 6-9 Operation Profile Maintenance Window



The `admin` profile is automatically assigned access to all domains. This operation profile should be assigned to users who require a complete picture of the monitored network. It is recommended that no access be deleted from this user's access domain. This guards against any part of the network being left unmonitored.

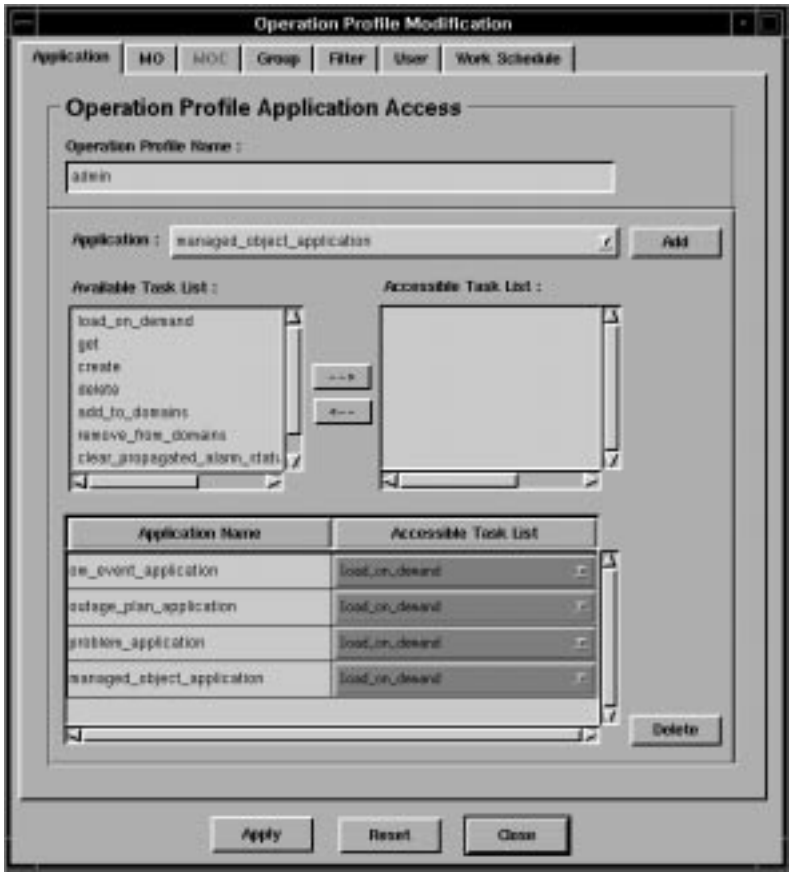
To create a new operation profile:

1. Enter a name for the operation profile in the `Operation Profile Name` text box.
2. Click **`[Add]`**.

   The `Operation Profile Modification` window shown in Figure 6-10 appears.

NOTE It is often faster to copy the `admin` profile and modify the copy to the desired settings than it is to create a new profile and assign all desired tasks.

Figure 6-10 Operation Profile Modification Window



This window has seven option tabs that enable association of access rights for the operation profile shown in the `Operation Profile Name` text box. The seven option tabs are used to associate the following information with the above named operation profile:

`Application` Applications and the tasks under them.

MO	Individual managed object domains.
MOC	Not used.
Group	Management domain groups.
Filter	Problem filters.
Users	Topology GUI users.
Work Schedule	Work schedule.

3. Save the operation profile by clicking **[Apply]** in the window. All additions, modifications, and deletions made in any of the tabs are saved.

Associating Applications with an Operation Profile

To associate an application with an operation profile, follow these steps:

1. Click the Application tab in the Operation Profile Modification window.
2. Select the application to be associated from the Application list. This list shows the applications configured with the Application:Application Maintenance function.

   There are four applications:

   - managed_object_application

     This is the application name associated with the actions taken with respect to managed objects via the managed object manager. This application opens the map presenter.

   - problem_application

     This is the application name associated with actions taken with respect to problems for managed objects via the problem manager. This application opens the problem presenter.

   - outage_plan_application

     This is the application name associated with actions taken with respect to outage plans for managed objects via the outage plan manager. This application opens the outage plan presenter.

   - om_event_application

     This is the application name associated with actions taken with respect to OM events for managed objects via the OM Event

manager. This applicationopens the OM event presenter.

For more information on the applications and associated tasks, see Appendix C , “Applications and Tasks.”

3. Select the tasks to be associated with this operation profile. Multiple selection is allowed.
4. Click **[-->]** to move the tasks to the `Accessible Task List` list.
5. Select any tasks that must be removed from the `Accessible Task List` list, and click **[<--]** to move the tasks back to the `Available Task List`.

   This actions makes the tasks unavailable to the current operation profile.
6. Click **[Add]**.

   This action copies the application name and the associated tasks to the lower list. The information is listed under the column heads Application Name and Accessible Tasks List.

Repeat steps 2 through 6, until all required applications and tasks have been associated with the operation profile being defined.

WARNING **Do NOT modify or delete the applications and their tasks under any circumstances.**

To remove any of the associated application and tasks lists:

1. Select the application name and task list row from the scroll box at the bottom of the window.
2. Click **[Delete]** to delete the application from the list.

   This removes the application' s association with the operation profile. However the application and the tasks are still available in the Operation Profile Configurator for use with other operation profiles.

Viewing the Tasks Under an Application To view the list of tasks associated with an application:

1. Select the name of the application from the `Available Applications` list.
2. Click **[Modify]**.

This action opens the Application Association window shown in Figure 6-11. The Available Tasks list shows the tasks associated with the selected application.

Figure 6-11 Application Association Window

Saving the Application Association Click **[Apply]** to apply the association. The association is immediately effective.

Click **[Reset]** to undo any changes made since the last **[Apply]**.

Click **[Close]** to close this window and return control to the Operation Profile Maintenance window.

Associating Management Domain Groups with an Operation Profile

To associate a management domain group with an operation profile:

1. Click the Group tab in the Operation Profile Modification window.

2. Select the management domain groups to be associated with the operation profile from the Available Management Domain Groups list. Multiple selections are allowed.
3. Click **[-->]** to move the selected management domain groups to the Accessible Management Domain Groups list.

   All management domain groups listed in the Accessible Management Domain Groups list are accessible to the operators linked to this operation profile being configured.

   Although many management domain groups are no longer listed in the Available Management Domain Groups list, they are still available to other operation profiles. A management domain group can belong to more than one operation profile.
4. Select any management domain groups that you do not wish to associate with this operation profile from the Accessible Management Domain Groups list, click **[<--]** to move them out of this list and into the Available Management Domain Groups list.

   This action makes the selected management domain groups unavailable to the current operation profile.

Saving the Management Domain Group Association Click **[Apply]** to save the association. The saved association is immediately effective.

Click **[Reset]** to undo any changes made since **[Apply]** was last selected.

Click **[Close]** to close this window and return control to the Operation Profile Maintenance window.

Associating Managed Object Domains with an Operation Profile

There may be some managed object domains, not covered by the management domain groups, that must be associated with the operation profile. The process of adding the managed object management domains to the operation profile is similar to that of adding the management domain groups.

To associate these managed object management domains individually to the operation profile:

1. Click the Managed Object tab in the Operation Profile Modification window.
2. Select the managed object management domains to be associated

with the operation profile from the Available MO Management Domains list. Multiple selection is allowed.

3. Click **[-->]** to move the selected managed object management domains to the Accessible Management Domains list.

   All managed object management domains listed in the Accessible Management Domains list are accessible to the operators linked to this operation profile.

   Although many managed object management domains are no longer listed in the Available MO Management Domains list, they are still available to other operation profiles. A managed object management domain can belong to more than one operation profile.

4. Select any managed object management domains that you do not wish to associate with this operation profile from the Accessible Management Domains list, and click **[<--]** to move them out of this list and into the Available MO Management Domains list.

   This action makes the selected managed object management domains unavailable to the current operation profile.

Saving the Managed Object Domain Association Click **[Apply]** to save the association. The window saved association is immediately effective.

Click **[Reset]** to undo any changes made after **[Apply]** was last selected.

Click **[Close]** to close this window and return control to the Operation Profile Maintenance window.

Associating Problem Filters with the Operation Profile

The process of adding problem filters to the operation profile is similar to that of adding the management domain groups and managed object management domains. To associate problem filters with the operation profile:

1. Click the Filter tab in the Operation Profile Modification window.
2. Select the problem filters to be associated with the operation profile from the Available Filters list. Multiple selections are allowed.
3. Click **[-->]** to move the selected problem filters to the Accessible Filters list.

The problem filters in the Accessible Filters list are used for the operators linked to this operation profile.

Although many problem filters are no longer listed in the Available Filters list, they are still available to other operation profiles. A problem filter can belong to more than one operation profile.

4. Select any problem filters that you do not wish to associate with this operation profile from the Accessible Filters list, and click **[<--]** to move them out of this list and into the Available Filters list.

   This actions makes the selected problem filter unavailable to the current operation profile.

Saving the Filter Association Click **[Apply]** to save the association. The saved association is immediately effective.

Click **[Reset]** to undo any changes made since **[Apply]** was last selected.

Click **[Close]** to close this window and return control to the Operation Profile Maintenance window.

Associating Problem Filters with an Operation Profile After you have created problem filters and operation profiles, you can associate operation profiles with a problem filter and you can view the operation profiles that are currently associated with a filter.

To associate operation profiles with problem filters:

1. Click **Filter:ProblemFilter Maintenance**.

   The Problem Filter Maintenance window appears.

2. Select the problem filter of interest from the Problem Filter Name list.

3. Click **[Modify]**.

   The Problem Filter Modification window displays.

4. Click the OperationProfile tab to view the associated operation profiles.

   The Available Operation Profiles list shows the operation profiles currently configured in the system. The Associated Operation Profiles list shows the operation profiles associated with the problem filter.

Associating Users with an Operation Profile

The process of associating users to the operation profile is similar to that of adding problem filters.

To associate users with the operation profile:

1. Click the User tab in the Operation Profile Modification window.
2. Select the user names to be associated with the operation profile from the Available Users list. These are OV user names created using the User Maintenance function.

   Multiple selection is allowed.
3. Click **[-->]** to move the selected user names to the Accessible Users list.

   The user names in the Accessible Users list are linked to this operation profile.

   Although many user names are no longer listed in the Available Users list, they are still available to other operation profiles. Users can linked to more than one operation profile.

   This User tab enables you to associate the currently opened operation profile with multiple users.

   To associate multiple operation profiles with a user, use the User:User Maintenance function.
4. Select any user names that you do not wish to associate with this operation profile from the Accessible Users list, and click **[<--]** to move them out of this list and into the Available Users list.

   This actions disassociates the user from the current operation profile.

Saving the User Association Click **[Apply]**. This action associates the users with the operation profile. So when the user logs in the system-default user Map Presenter map will contain the maps with operation profile names. The user can select any of the these maps to view the object symbols under them.

Click **[Reset]** to undo any changes made since **[Apply]** was last selected.

Click **[Close]** to close this window and return control to the Operation Profile Maintenance window.

Associating Work Schedules with an Operation Profile

Each operation profile can have a work schedule defined for it. The work schedule covers the following information:

- Whether there is a time restriction for operators using this profile. If there is a time restriction, the period during which this profile is accessible should be defined. *Operators accessing the topology GUI with this profile can log on and initiate any function only during the specified period.*
- If there is a time restriction, the time zone for the timing provided.

Define Work Schedule You can specify the following information regarding the operator login:

- Days of the week when the operator can log on.
- Time of the day when the operator can log on.
- The time zone associated with the above schedule.

Define the work schedule for the operation profile.

Table 6-5 describes the information needed to define a work schedule:

Table 6-5 Work Schedule

Parameter	Description
Operation Profile Name	This is the name of the operation profile for which the work schedule is being defined.
Time Zone	Indicates the time zone to which this work schedule relates.
Login Schedule	This time restriction for the operational profile is optional. You can restrict the access of the operators linked to this operation profile by specifying the day, and period within the day when this operation profile is operational. *Operator can trigger operations only during the specified period.* After the operator's scheduled time passes, the operator cannot initiate any actions. Specify the login schedule stating the following information: • Days of the week when this operation profile is operational. • The start time and the end time for the period during the day when this operation profile is operational.

Associating Work Schedule with Profiles To associate a work schedule with the operation profile:

1. Click the `Work Schedule` tab in the `Operation Profile Modification` window.
2. Select the `Time Restriction` option button.
3. For each time window, enter the start time and the end time by the day of the week and the time. Use the arrow keys beside the selection boxes to select the appropriate times.

   For each time period, click **[Add]**. The time window appears in the scroll box.

   If required, you can delete any time definition. Select the timing from the scroll box and click **[Delete]**.

   After defining the access time, save the work schedule.

NOTE

The operator can only log in during the period for which the operator is provided access. If the access time expires while the operator is logged in, the operator will not be able to make any updates through the GUI Client presenters.

4. Select the time zone from the `Timezone` list.

Saving the Work Schedule Details Click **[Apply]** to save the work schedule. This schedule is immediately effective.

Click **[Reset]** to undo any changes made since **[Apply]** was last selected.

Click **[Close]** to close this window and return control to the `Operation Profile Maintenance` window.

Saving the Operation Profile

To save the operation profile, click **[Apply]** in the window. All additions, modifications, and deletions made in any of the tabs are saved.

Click **[Reset]** to undo any changes made since the last **[Apply]**.

Click **[Close]** to close this window and return control to the `Operation Profile Maintenance` window.

Assigning Profiles to Users

To assign operation profiles to the operators, use the User:User Maintenance function or the OperationProfile:Operation Profile function of the Operation Profile Configurator to link the operator's user name (login ID) to the operation profiles.

Associating Operation Profiles with Users

After you have created operation profiles, you can associate operation profiles with users. You can also view the operation profiles that are currently associated with a user.

To associate operation profiles with a user:

1. Click **User:User Maintenance**.

   The User Maintenance window displays.

2. Select the user name of interest from the User Names list.

3. Click **[Modify]**.

   The User Modification window appears.

4. Click the OperationProfile tab.

   The Available OperationProfile list shows the operation profiles currently configured in the system. The Associated OperationProfile list shows the operation profiles associated with the OV User.

   This function is very useful when creating new users and maintaining them. You can select multiple profiles from the Available OperationProfile list and use **[-->]** to make them accessible to the selected OV user.

   The association is saved on clicking **[Apply]** and is immediately effective.

Customizing Topology GUI Login

To customize the topology GUI logon process add a topologin.<user> file or a topoprofile.<user> file in the system. Use the topologin.<user> file to log onto the topology GUI with a different user name than the OVO user login or to select an operation profile other than the default. Use the topoprofile.<user> file to log onto the topology GUI with a different operation profile than the default operation profile configured for that user.

Adding Authentication

By default, no additional login authentication is required by operators to launch the topology GUI from the OVO operator GUI. Administrators can add additional authentication by doing the following configuration steps:

- To add authentication, modify either the `guicstart` script on UNIX systems or the `%OVCAROOTDIR%\bin\start_ovcsa_gui.bat` script on NT systems to include:

  ```
  export FORCE_TELCO_GUI_LOGIN_AUTHENTICATION=yes
  ```

- Decide whether the authentication should be command line driven or via an interactive login GUI.

For command line authentication:

- Edit `%TEMP%/topologin.<`*`user`*`>` file to contain the topology GUI hostname, OS user name, OS password, and operation profile. Each of the entries should be included in the order specified and on a separate line.
- When the `topologin.<`*`user`*`>` file is present on the system, OV Topology Server reads this file to launch the topology GUI.

For an interactive login GUI authentication:

- On the OVO admin GUI, register a new application or change the executable of the existing `Launch UX GUI` function to `/opt/OEMF/V5.0/GUIC/oemf/util/login_telco_gui.ksh`. This displays the topology GUI login panel.
- On NT systems, register a new application or change the executable

of the existing `Launch NT GUI` function to `%OVCAROOTDIR%\bin\login_telco_gui.bat`. This displays the topology GUI login panel.

- Unregister any unwanted GUI launch applications.

NOTE When `FORCE_TELCO_GUI_LOGIN_AUTHENTICATION` is on, and neither `topologin.<user>` file is present nor the GUI launch application was changed, then authentication fails.

Changing Topology GUI profiles

By default, the topology GUI autoselects the first operation profile defined for that user. If an operator prefers to launch the topology GUI using a different operation profile, then edit the `%TEMP%/topoprofile.<user>` file. In this file, specify only the operation profile to be used when the topology GUI is launched from the OVO operator GUI.

7 Customizing the Topology GUI

The Admin Panel is a graphical user interface that a network administrator uses to set up operators' environment in the topology GUI. The Admin Panel provides the ability to define map access, define map object icons, and modify the menu selections, color mappings, and map display attributes. The Admin Panel can also be used to define and customize settings for different users and operation profiles.

Common Administrative Tasks

A network administrator can configure the operator environment by performing the following tasks:

- Assign operator profiles to operators.
- Create or modify symbols used to represent map objects.
- Create color mappings or bitmap mappings for problem, outage, and OM event tables.
- Define the mapping of blinking objects on the map presenter.
- Modify the operator's view by specifying presenter colors or fonts or setting the alarm bell.
- Define the visibility and location of menu commands.
- Define new actions and/or create menu commands to access those actions.

These tasks are described in this chapter.

Using the Admin Panel

The Admin Panel is a graphical utility used for day-to-day administration and customization of the user interface. The utility must be run by the network administrator (`oemfadm`) on the GUI Server machine. Operators do not have access to the Admin Panel.

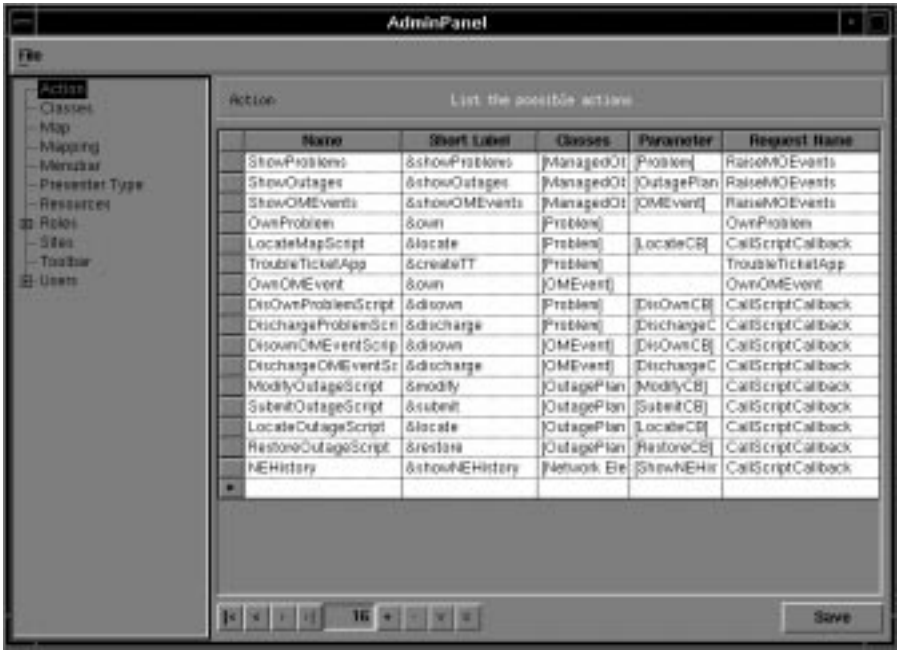
Starting the Admin Panel

To invoke the Admin Panel, log on as user `oemfadm` and execute:

`/opt/OEMF/V5.0/GUIS/oemf/util/guisadmin`

The Admin panel shown in Figure 7-1 appears.

Figure 7-1 Admin Panel



The Admin Panel is divided into two frames:

- The left frame lists the datatypes that can be selected for

customization in alphabetical order in a tree structure. The first datatype is `Action`, which is the default selected datatype when the Admin Panel is invoked.

- The right frame displays information that can be configured for the datatype selected in the left frame. The attributes displayed in the right frame depend on the selected datatype. In Figure 7-1, the datatype `Action` is selected and a list of pre-defined actions is displayed in the right frame.

Click a datatype in the left panel to select it. The right frame refreshes to display the details corresponding to the selected datatype.

WARNING **The following datatypes should not be customized with the Admin Panel:**

- **`Roles`. This datatype lists the profiles defined in the `Operation Profile Configurator`. The Roles must not be modified using the Admin Panel. All role definitions and modifications must be made using the `Operation Profile Configurator` only.**
- **`Users`. This datatype lists the users that have been defined using the `Operation Profile Configurator`. All User configuration must be made through the `Operation Profile Configurator` and not the Admin Panel.**

The `Presenter Type` datatype displays the presenter layout and dynamic view. This datatype is read-only, and no changes can be made through the Admin Panel.

Navigation Buttons are provided at the bottom of the right panel. These buttons can be used to move the cursor within the table in this panel:

[|<]	Moves to the first row.
[<]	Moves up one row.
[>]	Moves down one row.
[>|]	Moves to the last row.
[+]	Inserts a row.
[-]	Deletes the current row.
[V]	Validates the current changes.

[x]	deletes the current changes in the current row.
[Save]	saves the information entered in the panel.

NOTE To ensure that the changes made to the Admin Panel take affect, click [**Save**] before leaving any modified frame.

The number shown in the center of the navigation buttons indicates the number of entries in the displayed table.

Read-Only Information in the Admin Panel

The following information cannot be changed in the Admin Panel, but can be viewed.

Viewing the List of Users

The list of users who have access to the topology GUI are listed in the Users datatype of the Admin Panel. To view the list of users:

1. In the left frame:

   Click once on the plus symbol (+) associated with the `Users` datatype. The list of users appears in the left frame. To hide the list, click the minus symbol (-).

   The user names are listed as a branch of the tree under the datatype `Users`, as shown in Figure 7-2. Select a username.

2. In the right frame:

   All valid roles for the selected user are listed.

   The default administrator user, `oemfadm`, is created during installation and configuration.

Figure 7-2 Users Data Panel

Viewing the List of Roles

Roles (operation profiles) are sets of user access domains defined using the Operation Profile Configurator. Each profile validates the user to access one or more:

- Managed object domains
- Managed object class domains
- Application domains
- Problem filters

The profile could also define specific time periods during which a user can access applications. Each user can be associated with one or more profiles.

To view a list of defined profiles with the Admin Panel:

1. In the left frame:

   Click once on the plus symbol (+) associated with the `Roles` datatype. The list of roles appears in the left frame. To hide the list, click the minus symbol (-).

   The roles are listed as shown in Figure 7-3.

   Using this method, you can view the list of profiles while customizing

other datatypes in the `Admin Panel`.

2. In the right frame:

   To view the list of profiles in the right frame, click `Roles` in the left frame. All profiles created using the `Operation Profile Configurator` are listed as shown in Figure 7-3.

   The default administrator role, `admin`, is created during installation and configuration.

Figure 7-3 Roles Data Panel

Viewing the Presenter Types

The presenters are an integral part of OV Topology Server. A presenter is a window that displays map, table, or chart data and includes a defined set of selectable menu commands. The list of presenters is visible in the Admin Panel. The datatype `Presenter Type` displays the details of the presenters that have been configured.

To view the list of presenters:

1. Select the datatype `Presenter Type` from the left frame to display the configured presenters' attributes in the right frame.

   The datatype displays information about the presenter window. Figure 7-4 shows the datatype `Presenter Type` and the associated attributes:

Figure 7-4 Presenter Type Panel

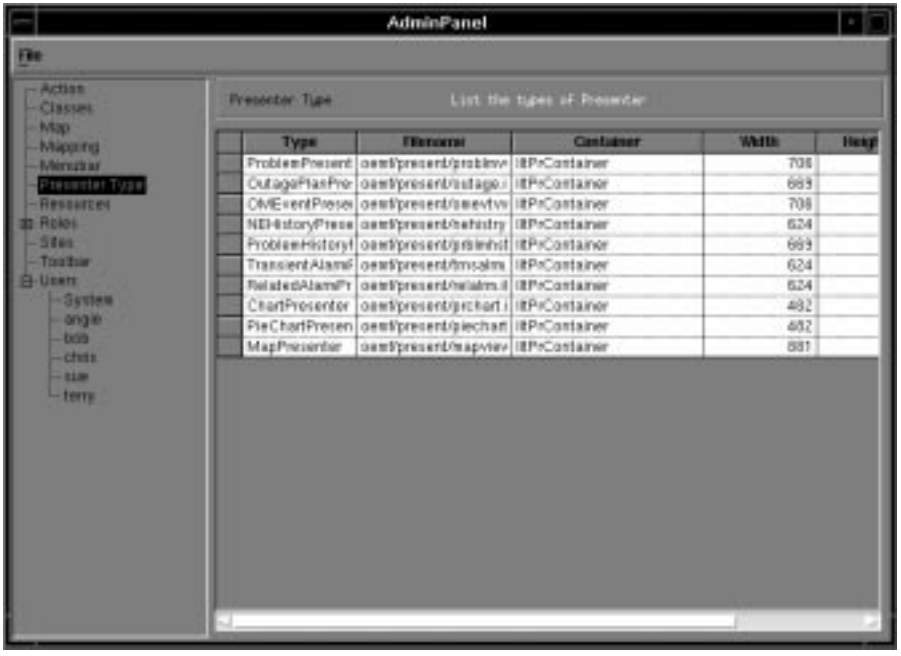


Table 7-1 describes the Presenter Type attributes:

Table 7-1 Attributes of Presenter Type

Column	Description
Type	Name of the presenter type
Filename	Pathname (relative to `/opt/OEMF/V5.0/ilog/oemf/data`) of the file that contains the presenter's panel description in `.ilv` format
Container	Name of the C++ container class instantiated for the presenter type
Width	Default width of the presenter window when it is invoked

Table 7-1 Attributes of Presenter Type

Column	Description
Height	Default height of the presenter window when it is invoked

Site Information

Select the `Sites` datatype to display the presentation database sites for the installation. The list of sites may be useful when resolving presentation database replication conflicts.

Change Ownership to Maps and Backgrounds

By default, maps created through the map presenter are available only to the user and role of the session when the maps are made. Maps created by the user `oemfadm` with the role `admin` are available as shared maps to all users. Maps created by the user `oemfadm` are available as shared maps to all users with the role of the session when the maps are made.

Changing Ownership of a Map

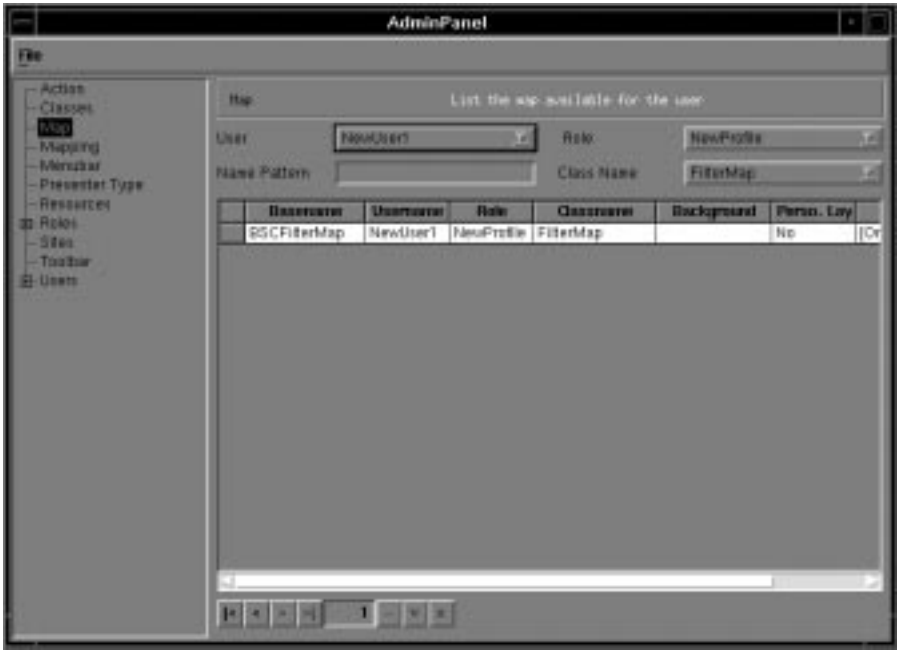
The availability of new maps to operators and roles depends on the operator and role that created the maps. To re-assign a map to a specific user and role:

1. Identify the map to change:
   - Specify the selection criteria for the `User` and `Role` to be the operator and role that created the map.
   - Specify the class name to be `Filter Map`.
   - Leave the `Name Pattern` blank.
2. In the row for the desired map, select a username from the list in the `Username` cell. To make this map available to all users, select `*`.
3. Select a role from the list in the `Role` cell. To make this map available to all roles, select `*`.
4. Click `[V]` to validate the selection.

   A message box appears: `The map already exists. Do you want to create a new map or update the current map?`
5. Click `[Modify]`.

   The map no longer appears in the Map datatype list because the new `user` and `role` no longer meet the criteria specified at the top of the panel.
6. At the top of the panel, change the selection criteria for the `User` and `Role` to match the user and role specified for the map in steps 2 and 3. The map is now listed as available for the newly specified user (or role).

Figure 7-5 Map Datatype Showing New Map Ownership



Copying a Map for a New User

When you specify the User and Role at the top of the Map datatype, all maps for the specified user and role are displayed. For example, if you select the User * and the Role *, all maps that are in the specified ClassName display. To copy a map that is currently displayed in the Map datatype panel for access by a new user or role:

1. Click the `User` cell, then select a username from the list.
2. Click the `Role` cell, then select a role from the list.
3. Click `[V]` to verify the entry.

   A message box appears: `Do you want to create a new map or update the current map?`
4. Click `[Create]` to copy the map for a new user and role.
5. At the top of the panel, change the selection criteria for the `User` and `Role` to specified in steps 1 and 2.

The copy of the map is now listed as available for the newly specified user (or role).

Adding Map Backgrounds

To specify a bitmap graphic to display as a background to a map in the map presenter, view the Map datatype in the Admin Panel. Enter the name of the background bitmap file (for example, `canada.gif`) in the Background cell for the map (see Figure 7-5). Click `[V]` to verify the entry.

Additional Map Information

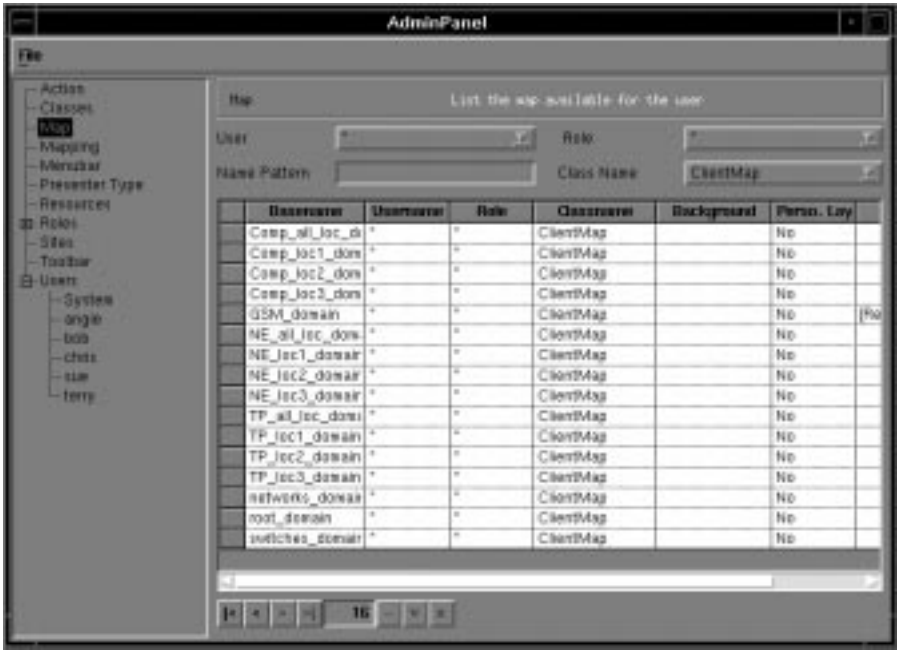
The `Map` datatype displays information about the maps available in the map presenter.

To view the list of maps:

1. Select the datatype `Map` from the left panel.

   The list of maps that have been configured for the map presenter appears in the right panel as shown in Figure 7-6.

Figure 7-6 Map Datatype Panel



Four criteria can limit the display in the table. Specify `User`, `Role`, and `Class Name` from the lists. `User` and `Role` define the user access to the map. `Class Name` defines the type of map. Enter a `Name Pattern` to limit the list of maps to those containing the `Name Pattern` entry.

The following table describes the columns in the `Map` panel:

Column	Description
Basename	Name of the map.
Username	Name of the user. This name is the same as that displayed in the `User` list. This field allows you to change the user of the selected map or to create a copy of the map for another user.
Role	The role identifier. This role is the same as that displayed in the `Role` list. This field allows you to change the role of the selected map, or to create a copy of the map for another role.

Column	Description
Classname	Class name of the map, for example, ClientMap or FilterMap.
Background	Background file of the map. The specified file appears as the map background in the Map Presenter. The background graphic file (*.gif) must be stored in: `/opt/OEMF/V5.0/GUIS/oemf/data/gif`
Perso. Layout	Specifies whether the map is a personal layout of a shared map. The values are yes or no. A personal layout contains user-defined information about object locations on the map and the Map Object Classes associated with the objects. The personal layout is only visible to a single user.When the user deletes the personal layout map, the shared map is automatically displayed to the user.
Property	Indicates the properties of the map such as the filter specification for a Filter based map. Click [...] to open a Parameters Dialog to specify the property values.

Define Map Symbols

The symbols that represent objects on the map are defined by assigning a Map Object Class to each Managed Object Class (MOC). A selection of map object classes are predefined based on predefined parent classes. You can create your own map object classes by deriving from the predefined parent classes as described later in this section.

Predefined Map Object Classes

The predefined map object classes for nodes derive from a single grandparent class, NeworkElementClass. The predefined map object classes are displayed in Figure 7-7 on page 197.

NOTE

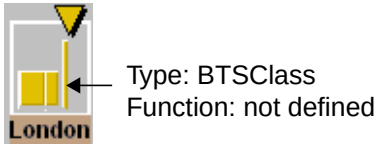
Each map object class is described by its parent class and three attributes that are defined in the Admin Panel: DisplayType, Type, and Function. Some of the attributes are predefined in the parent class, but can be reset in the Admin Panel. Example values for these attributes are shown in Table 7-4 on page 202.

The predefined map object classes that describe nodes are derived from the three parent classes described below:

- **SymbolicNEClass**—provides a square base. Color changes are displayed on the base shape. For example, the map object class displayed below is derived from the SymbolicNEClass with an additional function for the access icon. The icon (or function) appears in the lower left corner for normal size nodes and appears in the center of the object for small size nodes. The map object class displayed below is the predefined AccessClass:

Type: NE
Function: sym_access_type

- **PictorialNEClass**—provides an icon as the base shape. The base icon is defined by the Type attribute. A function can overlay on top of the base icon. Color changes are displayed on the icon itself. The map object class displayed below is the predefined BTSClass without any defined function:



- **ShapeNEClass**—provides a base shape. The predefined classes that derive from the ShapeNEClass are different shapes. If a function is specified, the icon appears in the center of the base shape. The map object class displayed below is the predefined MuxShapeClass:

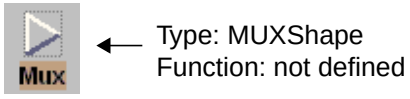
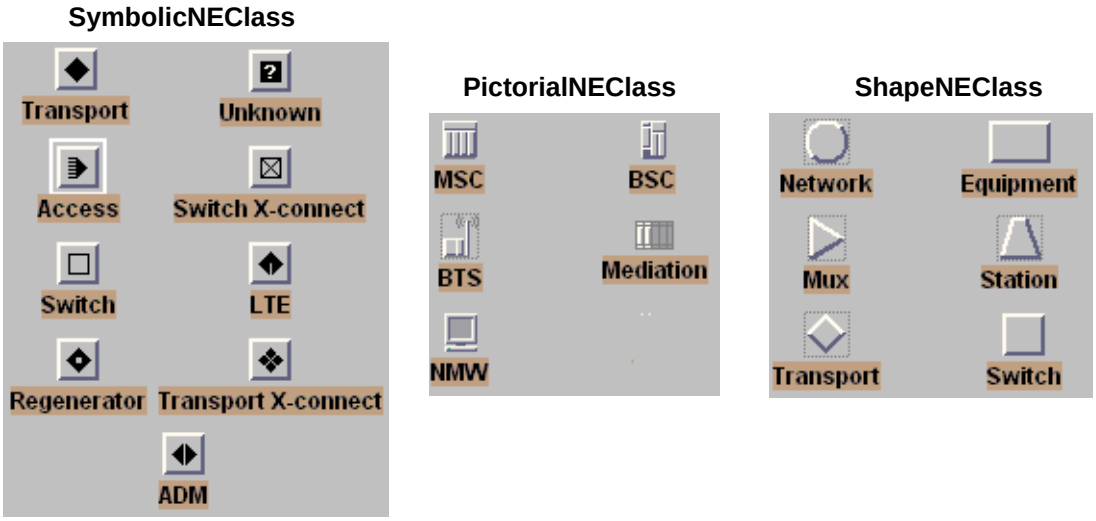
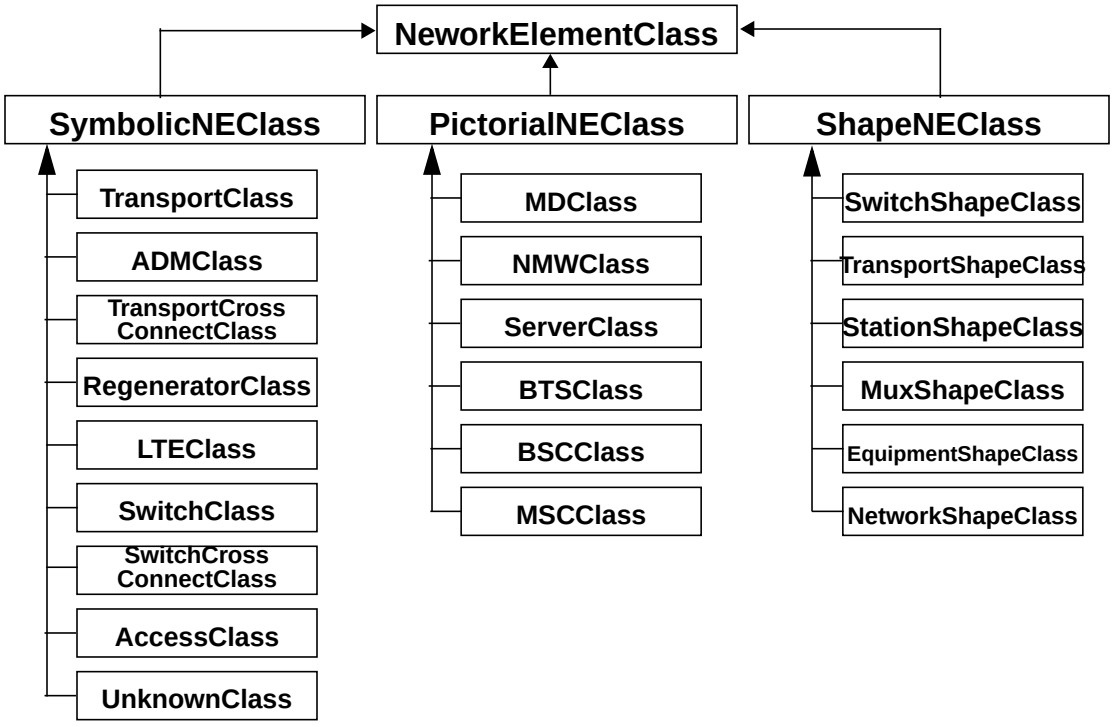


Figure 7-7 Graphic Images for Predefined Map Object Classes



The relationship between the predefined parent classes for the map object classes that describe map nodes is shown in Figure 7-8.

Figure 7-8 Predefined Map Object Classes



Five predefined display types associated with the highest level of parent classes are described in Table 7-2. Thus far, we have only discussed standard nodes that all derive from the DisplayType StandardNode. The additional DisplayTypes are used for group objects and off page connectors.

Although all five of the classes described in Table 7-2 could potentially be used as parent classes for user-defined map object classes, only the StandardNode is commonly used for the definition of standard node map object classes.

Table 7-2 DisplayType Attributes

DisplayType	Enum	Description
StandardNode	0	Used by NetworkElementClass. Base shape that can display relief (flat or dimensional) and color. Color is used to display status. Icons can be displayed on top of the square base shape to indicate the type of node.
PolygonalGroup	1	Used by PolygonalGroupClass. Polygonal base shape that can be reshaped to form any polygonal container object. This class is not usually used as a parent class when creating new map object classes.
RectangleGroup	2	Used by RectangularGroupClass. Rectangular base shape that can be resized to form a rectangular container object. This class is not usually used as a parent class when creating new map object classes.
LinearGroup	3	Used by LinearGroupClass. Linear base shape that can be resized to form a linear container object. This class is not usually used as a parent class when creating new map object classes.
OffPageConnector	4	Used by the OffPageConnectorClass. Offpage connector base. This class is not usually used as a parent class when creating new map object classes.

User-Defined Map Object Classes

Additional map object classes can be created beyond those that are predefined by the system. A user-defined standard node map object class typically derives from the SymbolicNEClass, PictorialNEClass, or ShapeNEClass.

NOTE Do NOT use the NetworkElementClass as a parent class.

The three map object class attributes may be specified, as necessary:

Table 7-3 Attributes to Describe Map Object Classes

Name	Type	Value
Type	Static Only supported value.	NE is used to obtain a square base shape. All classes that derive from SymbolicNEClass inherit the Type NE for standard network element. Other possible values are: For PictorialNEClass—MD, NMW, BTS, BSC, or MSC. For ShapeNEClass—SwitchShape, TransportShape, StationShape, MuxShape, EquipmentShape, or NetworkShape. For Map Links—Fiber, Electrical, or CNET.
DisplayType	Static	Since the parent class usually defines this value, no entry is required. The DisplayType for a standard node is 0. All classes that derive from NetworkElementClass inherit the DisplayType of 0. Other values are described in Table 7-2.

Table 7-3 Attributes to Describe Map Object Classes

Name	Type	Value
Function	Static	Either: *NewFunctionName* OR enter an existing value from the Mapping datatype for the user listed in the FunctionToIconMapping list (see below). The Function value is a logical name for the graphic icon. The Function value appears when the cursor moves over the icon in the Map Presenter. The predefined functions include—Transport, ADM, TransportCrossConnect, Regenerator, LTE, Switch, SwitchCrossConnect, Access, or Unknown. This value is only required for an icon to display on top of the base element.

NOTE Because the function is defined as a bitmap icon to be displayed on top of the base shape, the icon must be small enough so as not to obscure information conveyed by the base shape such as severity status. A typical icon size for a square symbol is 38x38 pixels. The icon (mapped to the function) is specified by a `.gif` file defined in the `Mapping` datatype of the Admin Panel in the `FunctionToIconMapping`.

The relief of the object (indicating whether the map object class has an associated object) and the color of the object (indicating the calculated severity status) are not set as attributes, but are instead determined by the network data.

Two utilities are provided to create new map object classes, the Admin Panel and the `guidbmocsym` utility, that accesses a text-based file. Both methods are described here.

Using the Admin Panel to Create Map Object Classes

To use the Admin Panel to create a new map object class with the name `new_func_demo` and a symbol that is a square base shape with a new icon in the lower left corner:

1. Click the `Classes` datatype in the Admin Panel to define the Parent Class for the new map object class.
2. At the end of the `Name` column, enter a name for the new map object class. For example, enter **`new_func_demo`**
3. In the `ParentClass` column, enter a Parent Class for the new map object class (see Figure 7-8 for a list of valid predefined Parent Classes). For example, enter **`TransportClass`**.

NOTE

In this example, the TransportClass is specified as the parent class and the value of the function is modified. An alternate procedure is to specify the SymbolicNEClass as the parent class and specify the original value for the function.

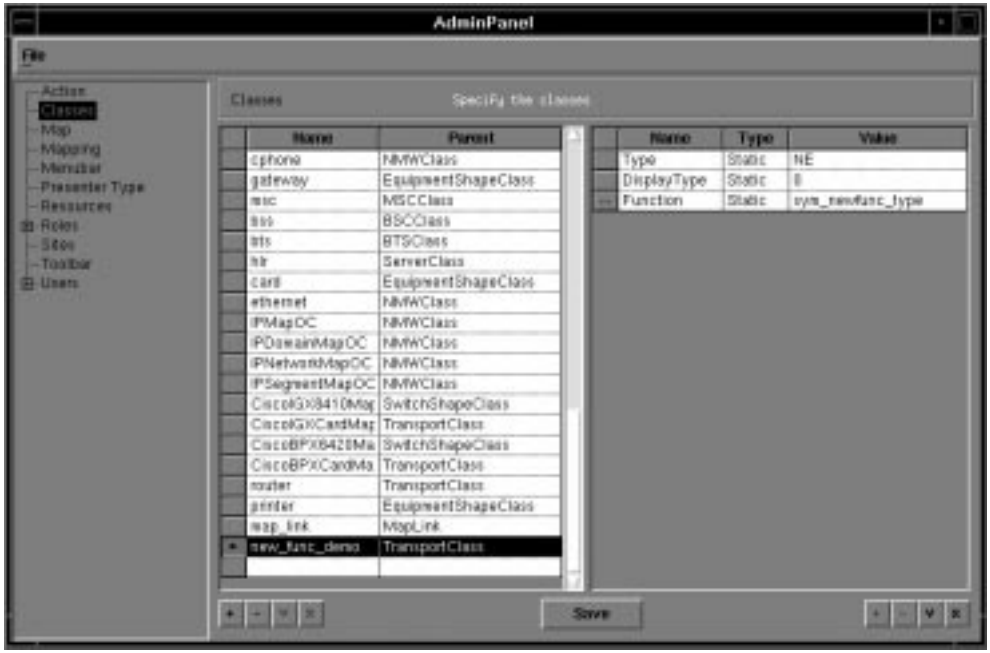
4. Select the row that contains the new map object class to bring up additional information in the right side of the `Classes` datatype panel. Enter the following information for the new map object class. (See Table 7-5 on page 203 for additional information about the fields in the Classes datatype.)

Table 7-4 Attributes for the Map Object Class

Name	Type	Value
Type	Static	NE
DisplayType	Static	0
Function	Static	sym_newfunc_type

The final display in the Classes datatype is shown in Figure 7-9.

Figure 7-9 Complete Class Specification



The fields defined in the Classes datatype are described in Table 7-5:

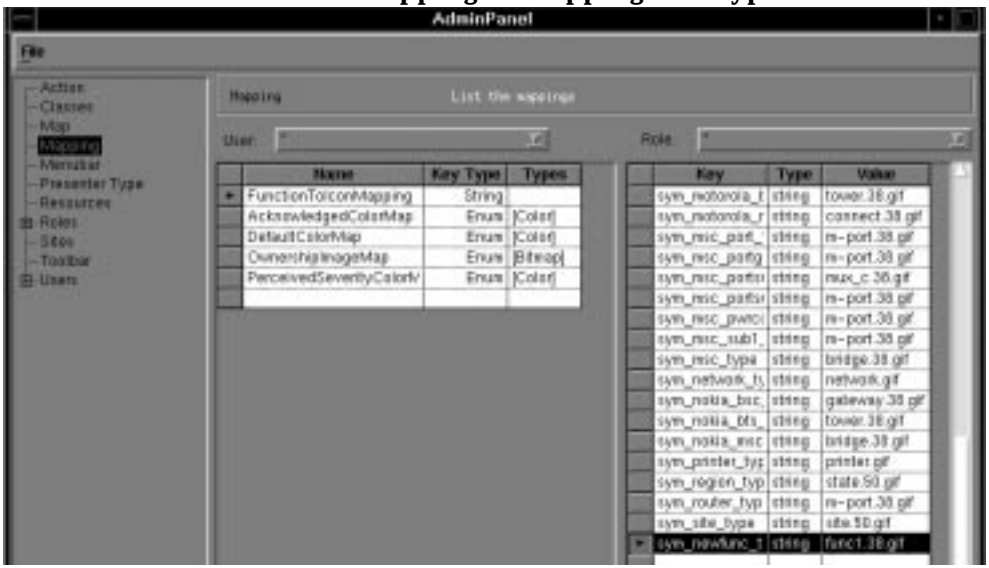
Table 7-5 Classes Datatype Fields

Column	Description
Name	Map object class class name
Parent	Name of the parent class for the new map object class. The predefined parent classes are listed in Figure 7-8, “Predefined Map Object Classes.”
Name	Name of the attribute (`Type`, `DisplayType` or `Function`). The name must be unique in its class.
Type	Indicates whether the attribute is `static` or `instance`. Table and Map Presenters only use the attribute, `static`. Static indicates that all users and roles use the same Type value.
Value	The values for Type, DisplayType, and Function are described in Table 7-3 on page 200.

5. The Function must be associated with a graphic icon for the map object class. Associate the Function with the icon in the function-to-icon mapping.

   In the Admin Panel, select the `Mapping` datatype for the user * and role *, then select `FunctionToIconMapping` to display the list of function names and their associated graphic icons as shown in Figure 7-10.

Figure 7-10 Function to Icon Mapping in Mapping Datatype



Using the Admin Panel to Map Map Object Classes to MOCs

To associate the map object class with the appropriate managed object class, in the `Mapping` datatype, select the `System` user and role *, then select `IltCoMo2MapObjClassMapping`.

- For a 1:1 relationship between MOC and map object class, enter:

  Key: *`MOC`*, Type: `String`, and Value: *`MapObjCls`*

- For a 1:Many relationship between MOC and map object class, enter:

  Key: *`MOC`*, Type: `Array`, and Value: *`[MapObjCls, MapObjCls, MapObjCls...]`*

When a single MOC is associated with multiple map object class values, the first value in the array is the default map object class. The additional

map object class values can be used for the MOC but must be explicitly assigned to the managed object instances.

Enter the MOC exactly as it is defined in the telecom configurator. Enter the map object class exactly as it is defined in the `Classes` datatype.

Using guidbmocsym to Create New Map Object Classes

The `guidbmocsym` utility imports the necessary information to create user-defined map object classes and define the mapping between MOCs and map object classes from a data file. This procedure is more efficient than using the Admin Panel to define many map object classes.

NOTE With the `guidbmocsym` utility, the map object class assignments are created for all users and roles. To create map object class to MOC mappings for different users and roles, use the Admin Panel.

Add new entries to the `.dat` file in `/etc/opt/OEMF/V5.0/GUIDB/share/conf`. These entries must contain information that defines the map object class and its mapping to the appropriate MOC. For example, the following lines define four map object classes:

```
< NokiaMSC nokia_msc MSCClass MSC StandardNode bridge.38.gif >
< MotorolaMSC motorola_msc MSCClass MSC StandardNode connect.38.gif >
< NokiaBSC nokia_bsc BSCClass BSC StandardNode gateway.38.gif >
< MotorolaBSC motorola_bsc BSCClass BSC StandardNode tower.38.gif >
```

The file contains the fields:

<MOC MapOC ParentClass Type DisplayType Function>

The icon files must be stored in `/opt/OEMF/V5.0/GUIS/oemf/data/gif`.

To import the data from the `.dat` file, execute the command:

/opt/OEMF/V5.0/GUIDB/bin/guidbmocsym -typeFile /
/etc/opt/OEMF/V5.0/GUIDB/share/conf/tgf_types.dat -create *NewData*.dat

NOTE The `tgf_types.dat` file describes the predefined map object classes and should not be modified.

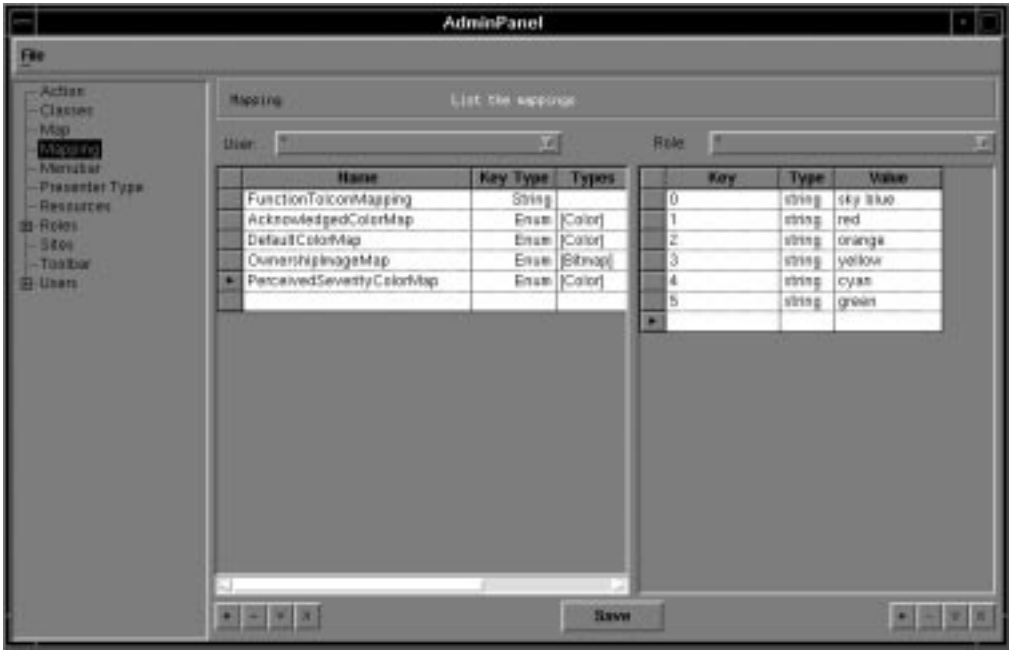
Create Mappings for Colors or Bitmaps

The `Mapping` datatype is used to make one-to-one associations between two lists of values. For example, the numbers for the severity levels in OV Topology Server can be mapped to the colors in the problem presenter; the palette of colors available and the colors used to indicate different states can be mapped to the state values.

`Mapping` can be defined for roles and users. By default, two sets of mappings are defined—one for the system defaults and the other for all users (*).

When you select the `Mapping` datatype, the right frame displays the `List of Mappings` as shown in Figure 7-11.

Figure 7-11 Mappings Panel



This right frame contains two columns.

In the left column of the right frame, the mapping elements are defined by:

Column	Description
Name	Name of the mapping element.
Key Type	Defined as either a `string` or an `enum` type key.
Types	Click in the cell, then click [...] to bring up the Parameters Dialog. In the Parameters Dialog, select a type from the Type list and enter a value of: Color, Bitmap, or Font.

The second column of the right frame lists the `Key`, `Type` and `Value` for the selected mapping element. First select a Type from the list. The Key fills in automatically.

All mappings are defined for user and role. To make the mapping available to all users and roles, Select '*' from the users and roles lists.

A mapping can be used with more than one type. For example, you can create a mapping between an alarm level and a color mapping. The mapping types are used in the `Options` panel of the problem presenter. For details, see the online help installed with the topology GUI.

System Mapping

A set of system level mappings that define the representation on the GUI Client is provided. None of the existing data in these mappings should be modified. You may extend the mappings in IltCoMo2MapObjClassMapping. This set consists of:

- IltShPropagateStateModelToTGOMapping

  Maps the OV Topology Server propagation state to the map objects.

- IltCoMo2MapObjClassMapping

  Maps the map object class to the MOC to assign symbols to objects in the map presenter.

- IltPrTablePresenterTypeMapping

  Maps the table presenter name to the table presenter function.

- MapClass2PresenterTypeMapping

  Maps the type of map class to the map presenter function.

- IltShAlarmCountModelToTGOSeverityMapping

  Maps the alarm severity levels to display the severity level.

- IltCoMapManagerProperties

  Maps the function names to the functions in the presenter.

Color Mapping

For all users (*), color mapping is available for customization of the topology GUI presenters. The mapping configured by default for all users includes:

- `DefaultColorMap`: used to list the default colors for the presenters.
- `AcknowledgedColorMap`: used to define the color mapping for the acknowledged and unacknowledged alarms.
- `PerceivedSeverityColorMap`: used to define the color mapping of the severity levels.

Mapping Bitmaps

Attributes in the topology GUI can be mapped to bitmaps as well as colors. To map bitmaps, include the bitmap files in the directory, `/opt/OEMF/V5.0/GUIS/oemf/data/gif` on the topology GUI and GUI Server machines, then complete the same procedure as for color mapping except choose the `Type` to be `Bitmap` and the `Values` to be the bitmap files. For example:

`OwnershipImageMap`: used to map bitmap images to indicate owned or unowned problems.

Changing the Perceived Severity Color Mapping

The Perceived Severity level assigned to each problem in OV Topology Server can be mapped to a color that appears in the problems presenter. A default color map is configured for the problems presenter. To change the default colors defined for the Perceived Severity Color Map:

1. In the Mapping datatype of the Admin Panel, select the `PerceivedSeverityColorMap` row.
2. In the right column the `Key` contains the enum for the Perceived Severity and its corresponding `Value` defines the color associated with the `Key`. Edit the colors in the `Value` cells using any values from the `Default Color Map`. The default values are shown in Table 7-6

Table 7-6 Sample Labels for Default PerceivedSeverityColorMap

Display Color	Severity	PerceivedSeverityEnum
Sky Blue	Indeterminate	0
Red	Critical	1
Orange	Major	2
Yellow	Minor	3
Cyan	Warning	4
Green	Cleared	5

3. Click `[V]` to validate the values.
4. Click `[Save]` to save the changes to the color map.

   The new color map is applied when an operator selects the `PerceivedSeverityColorMap` for use in the problems presenter.

Adding a New Color Map

For example, to create a new color map for an attribute in the Outage Plan Presenter, follow the instructions below. This example creates a color map for the Administrative State of the problem in the Mapping datatype panel.

1. In the left column of the mappings list, enter the following values:

   In the Name cell, enter **AdminStateColorMap**.

   In the Key Type cell, select String from the list.

2. Click in the Types cell, then click [...].

   The Parameters Dialog appears.

3. In the Parameters Dialog, select string from the Type list, then enter **Color** in the Value cell.

4. In the Parameters Dialog, click [V] to validate, then click [Apply].

5. In the Admin Panel, click [V] to validate the row.

6. Click to the left of the row to select it.

7. In the right column of the mappings list, click in the Key cell, and enter one of the administrative states listed in Table 7-7.

8. Select string from the Type list.

9. In the Value cell, enter the color to be mapped to the administrative state specified in step 7.

10. Click [V].

    A new row appears.

11. Repeat steps 7 through 10 for each value. (For the AdminStateColorMap, three rows must be completed for the three administrative states as described in Table 7-7.)

Table 7-7 Admin State Numerical Values

Administrative State (for String type)	Numerical Value (for Enum type)	Type	Color
Locked	0	string	Red
Unlocked	1	string	White
Shutting Down	2	string	Yellow

12. Click [Save].

    The AdminStateColorMap is listed in the Options Panel of the outage plan presenter.

Using Bitmap Mappings in the Topology GUI

In the Admin Panel, mappings can be established to map attribute values to bitmaps (or colors). Use the Options Panel of the presenters to apply these mappings, which are applied to the data in the table presenters. For example, the OwnershipImageMap sets up icons to represent whether a problem has been acknowledged.

The OwnershipImageMap is predefined in the Admin Panel to map check marks when the acknowledged value is 1 and x letters when the acknowledged value is 0. From the Admin Panel, the bitmaps used to map to these values can be changed. Similarly, other mappings that map bitmaps to values can be created.

For example, to create a bitmap mapping to Admin State, follow a similar procedure to that described in “Adding a New Color Map” on page 210:

1. Create three bitmaps for the three Admin State values. Be sure the files for the bitmaps are in the appropriate directory on the topology GUI machine:

   - Windows NT operating system:
     `Program Files\HP OVCA 1.02\data\gif`
   - UNIX systems: `/opt/OEMF/V5.0/ilog/oemf/data/gif`

2. In the left column of the mappings list, enter the following values:

   In the Name cell, enter **AdminStateBitmapMap**.

   In the Key Type cell, select enum from the list.

3. Click in the Types cell, then click [...].

   The Parameters Dialog appears.

4. In the Parameters Dialog, select string from the Type list, and enter **Bitmap** in the Value cell.

5. In the Parameters Dialog, click [V] to validate, and click [Apply].

6. In the right column of the mappings list, select string from the Type list.

   The Key number fills in automatically. Be sure that the key matches the filename appropriates. (For example, shuttingdown.gif should match the Key of 2.)

7. In the Value cell, enter the bitmap filename that you want to map to the numerical value.

8. Click [V].

   A new row appears.

9. Repeat steps 6 through 8 for each numerical value. (For the AdminStateBitmapMap, three rows must be completed for the three Admin States as described.)

10. Click [V], then click [Save].

    The AdminStateBitmapMap becomes available in the Options Panel of the table presenters.

Configuring Local Form Probable Cause Mapping

Event messages that are received by the Mediation Device (MD) and Fault Management (FM) Server are mapped into CMISE format and processed. For greater flexibility, local form probable causes are supported as well as global probable cause values. The local form probable cause are mapped to enums that must be mapped to the appropriate string values for use in message processing. These string values, rather than the enums, display in the GUI Client presenters and are readable by the operators.

To map the local form probable causes from integers to their string equivalents:

1. Select the `Mapping` datatype.
2. Create a new mapping with the Name **`ProbableCauseLocalEnum`** (this Name must be used exactly as written here).
3. Specify the KeyType to be `enum`.
4. Leave the `Types` field blank.
5. Click `[v]` to validate the mapping entry.
6. In the right table, enter the enums and string equivalents, for example:

Table 7-8 Local Form Probable Cause Example

Key	Type	Value
0	string	ExceedThreshold
1	string	PowerProblem
2	string	SignalTooHigh

7. Click `[v]` to validate each entry as they are entered.

Local probable cause codes may not be sequential (i.e. 7, 15, 20, 28, and etc.). The mapping table requires entries to be entered starting with 0 and following the enumeration values in sequence (i.e. 0, 1, 2, 3, 4, and etc.). Thus, defining the string mapping values for the required local probable cause codes will require entries to be added to the mapping table that may not be needed. For example, if a local probable cause code of 5 needs to be mapped to a particular probable cause string, then the mapping table will look like the following.

Table 7-9 Local Probable Cause Codes Not Needed

Key	Type	Value
0	string	<blank>
1	string	<blank>
2	string	<blank>
3	string	<blank>
4	string	<blank>
5	string	ThresholdExceeded

When local form probable cause values are used to describe the attributes of a problem, the string value appears in the GUI Client presenters.

Configure Blinking Objects

Map objects can be configured to blink once per second when they have associated problems of a specified severity level and above. Blinking is only initiated for objects with unowned problems.

Blinking is initiated for unowned problems when the perceived severity increases above the specified threshold value or above the current value if no threshold is set.

Blinking stops when:

- All problems are owned.
- The blink expiry time elapses without new problems arising on the object that meet the perceived severity requirements or existing problems updating to a higher perceived severity.

If the perceived severity of the problem increases or new problems arise that meet the perceived severity requirements during the blink time interval, the blinking timer resets the blink expiry time.

Blinking is only dependent upon changes to perceived severity and ownership of problems. No other problem attributes affect blinking.

This functionality is configured using the `Mapping` datatype in the `Admin Panel`. To enable this functionality, execute:

Step 1. Log on as `oemfadm` to the topology server machine and invoke the `Admin Panel`.

Step 2. Select the `Mapping` datatype from the left frame.

Step 3. Select the relevant users from the `User` list. (Select `*` to configure the object blinking function for all users.)

Step 4. From the `Role` list, select the role for which the objects must be configured to blink. (Select `*` to configure the object blinking function globally.)

Step 5. In the left column, for the `Name`, enter **`OVCSABlinkConfigMapping`**.

Step 6. In the `Key Type` cell, select `String` from the list.

Step 7. Click `[V]` to validate the entries.

Step 8. For the values of the mapping element **OVCSABlinkConfigMapping**, enter the values shown in Table 7-10 in the second column.

Click [V] to validate each row an add an empty row.

Table 7-10 Values for the Mapping Element OVCSABlinkConfigMapping

Key	Type	Value
EnableBlinking	**Boolean**	**Yes**
BlinkSeverity Threshold	**integer**	**blinkseveritythreshold** ***number*** (optional)

The Value for the Key **blinkseveritythreshold** is the integer value of the threshold severity level. For example, **2** sets blinking on for unowned problems with a severity of major or greater. This value is the severity value equal to or above which objects should blink. If no value is entered for the threshold, any new unowned problem initiates blinking.

Step 9. Click [Save] to save the entry.

Configure Blink Expiry Time

The blink expiry time is set in the Resources datatype. See page 217 for more information about the Resources datatype.

To set the blink expiry time:

1. Select the Resources datatype in the Admin Panel.
2. Specify the user and role for this expiry time.
3. Select the Map Presenter for Presenter Type.
4. In a blank row in the right frame of the Resources data type, enter:

   Gadget Type: **IltPrGrapherGadget**

   Gadget Name: ***

   Resource Name: **BlinkExpiryTime**

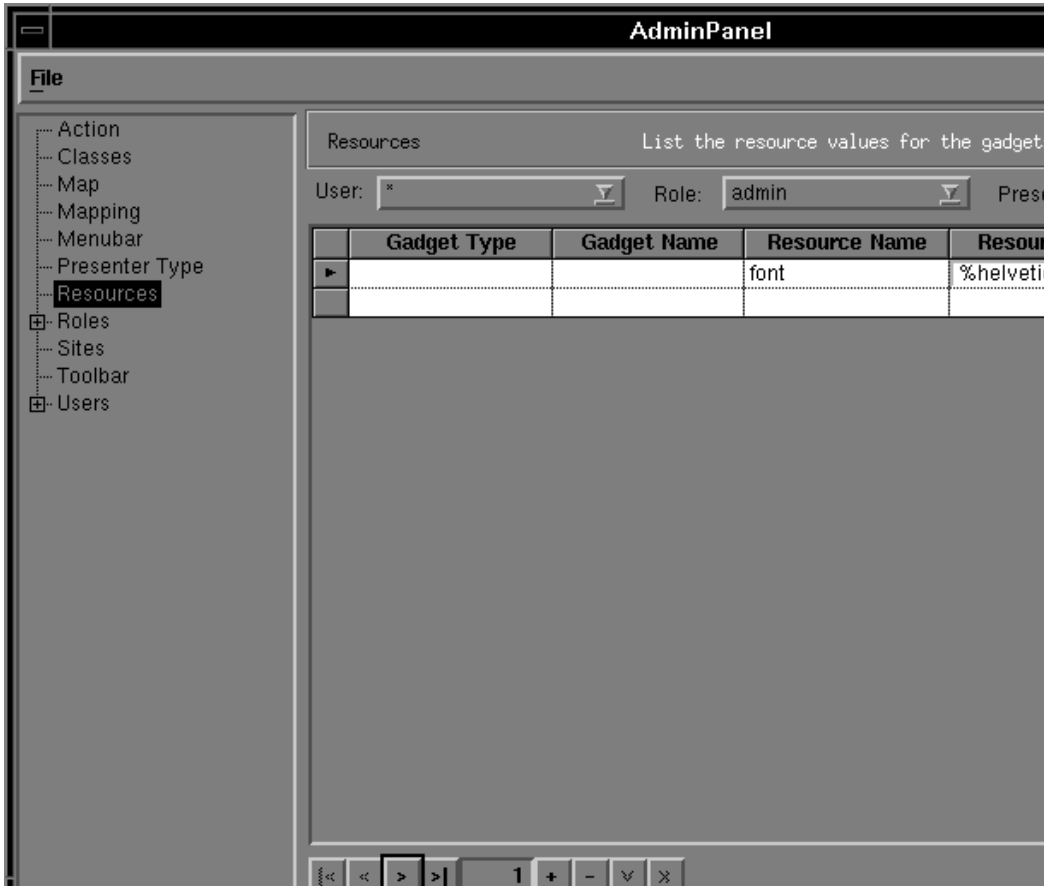
   Resource Value: **x** (the number of milliseconds that blinking endures)
5. Click [V], then [Save]. After you have saved this data, it cannot be modified. To change the expiry time, repeat these entries on an empty line in the Resources datatype.

Set Colors, Fonts, and Alarm Bell for the GUI Client

The Resources datatype is used to define colors, fonts, and alarm bells used in the topology GUI. Resources can be defined by role, user, and presenter type. At installation, resources are configured for the Map, Problems, Chart, and Pie Chart Presenter.

The Resources panel is shown in Figure 7-12.

Figure 7-12 Resources Panel



The following table defines the columns in the `Resources` datatype:

Column	Description
Gadget Type	Type of gadget for which this resource is applicable.
Gadget Name	Name of the gadget used.
Resource Name	Name of the resource, selected from the list. Three resource names are used for the alarm bell: `BeepOnNew`, `BeepOnIncrSev`, `BeepOnUpdated`. Specify either `background` or `foreground` as the resource name to access the `Color Chooser` panel. Refer below to the section “Defining Colors” for more information. Specify `font` as the resource name to access the `Font Chooser` panel. Refer to section “Defining Fonts” on page 220 for more information.
Resource Value	Value of the resource. The value depends on the type of resource specified in the previous column.

To modify the colors or fonts in the presenters:

1. Select the User, Role, and Presenter Type above the table to define where these changes will be effective.
2. Click in the ResourceName cell to specify background, foreground, or font.
3. Click in the ResourceValue cell to bring up the Font Chooser or Color Chooser dialog box. For details, see below.

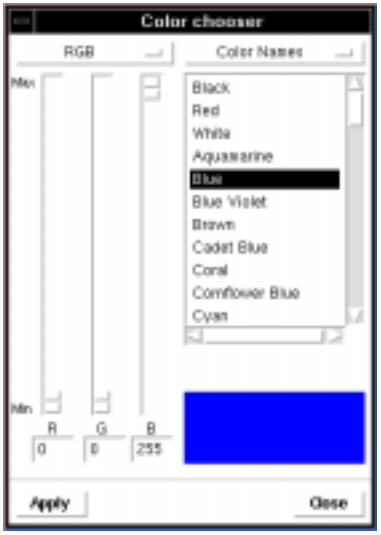
   The User, Role, and Presenter Type will automatically populate in the table based on the filters specified above the table.

Defining Colors

You can define colors for resources `background` and `foreground` using the `Color Chooser` panel. This panel is invoked from the `Resource Name` column. To invoke the `Color Chooser` panel:

1. In the `Resource Name` column, select `background` or `foreground` from the list.
2. Click in the `Resource Value` column, then click `[...]` to invoke the `Color Chooser` panel, as shown in Figure 7-13.

Figure 7-13 Color Chooser Panel



3. Click the `Color Name` in the standard palette. Customize the color by changing the RGB or HVS values with the slider bars.

   A sample color box, displayed below the `Color Name` palette, shows the new color and intensity.
4. Click `[Apply]` to confirm the color.
5. Click `[Close]` to close this window and return control to the `Admin Panel` window.

NOTE

In the table presenters, colors can be defined for the cells and headers by a network administrator with the Admin Panel by entering the following data in the `Resources` datatype:

The User, Role, and PresenterType are automatically completed based on the filter criteria entered in the text fields at the top of the window.

Enter: `Gadget Type`: **`IltPrTableGadget`**, `GadgetName`: **`*`**, `ResourceName`: **`cellBackground`** or **`cellForeground`** or **`Background`** or **`Foreground`**, `ResourceValue`: *`any defined color`*.

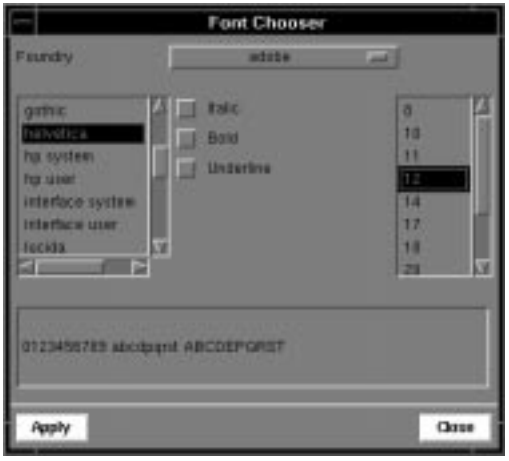
Since the `ResourceNames` cellBackground and cellForeground are not listed in the list, enter the name from the keyboard, then click <**`Enter`**>.

Defining Fonts

To define the fonts to be used within a gadget:

1. In the `Resource Name` column, select `font` from the list.
2. Click in the `Resource Value` column, then click `[...]` to invoke the `Font Chooser` panel, as shown in Figure 7-14

Figure 7-14 Font Chooser Panel



3. Set the font by name, style, and point size.

The lower panel provides a preview of the selected font.

4. Click [Apply] to confirm the font.
5. Click [Close] to close this window and return control to the Admin Panel window.

Defining Fonts

When colors and fonts are defined using the Resources datatype, the colors and fonts affect all the windows in the presenter. In contrast, fonts can be defined in the table presenters that affect only the headings and cells of the table.

The fonts for the header and table can be selected separately to help to distinguish regions of the table.

For information on setting the font for the table headings and cells, see the online help installed with the topology server.

Setting Alarm Bell

The alarm bell can be set by a network administrator with the Admin Panel or by the operator in the Set Alarm Alert panel. To set the alarm bell for a specified user and role:

1. Select the User and Role, then specify the Presenter Type to be Problems Presenter above the table to define where these changes will be effective.

   Three ResourceName cells are available: BeepOnNew, BeepOnIncrSev, and BeepOnUpdated.

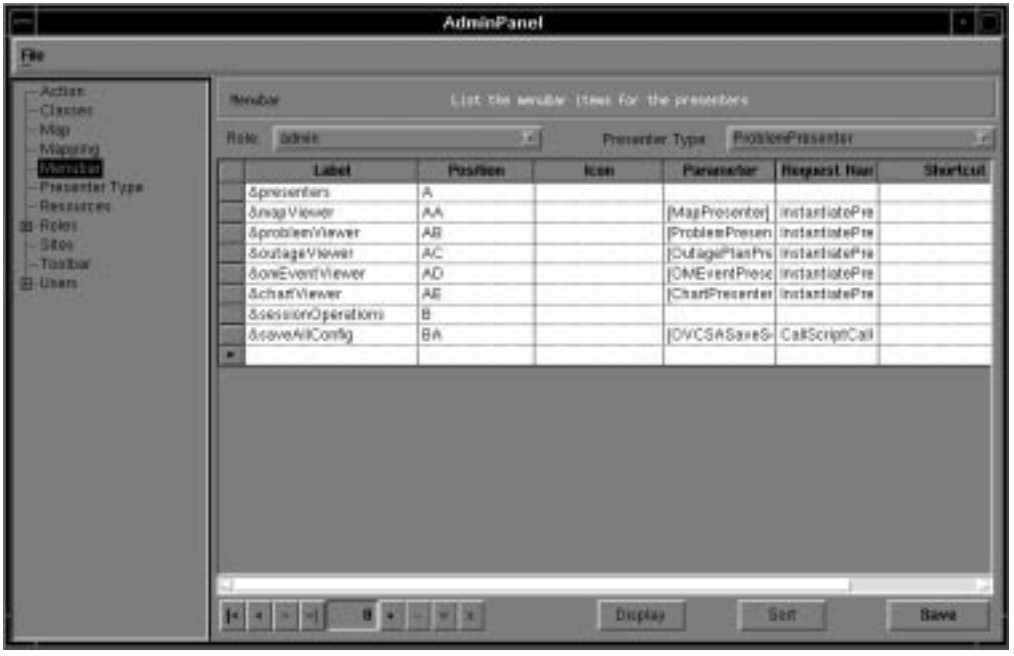
2. For each of the three ResourceNames, specify the ResourceValue cell to be True or False. True indicates that the bell sounds for an event.

   For example, True for BeepOnNew indicates that the alarm sounds when a new problem enters the Problems Presenter; True on BeepOnIncrSev indicates that the alarm sounds when the severity of a problem increases; True for BeepOnUpdated indicates that the alarm sounds when the problem is updated.

Define Menubars

Using the `Menubars` datatype, you can define menus, submenus, and commands, customized by role and presenter type for the presenter windows. All users view the same menubar. The lists at the top of the panel are used to specify which roles and which presenters will display the associated menu commands.

Figure 7-15 Menubars Panel



During installation, the menubars are defined for the roles, * and admin for several presenters.

To add new commands to the menubar, add a new row to the menubar table. The following table describes the information to be specified for each menu command.

Column	Description
Label	The menu name, as it will be displayed in the presenter window or use a localizable string for string substitution. The label can be localized if you use an **`&string`** for the label field in the Admin Panel, then store the actual label in the `/opt/OEMF/V5.0/ilog/oemf/data/oemf/present/oemf_msg.dbm` message catalog.
Position	Used to define the position of the menu command. For information on setting position, see “Defining the Menu Positions” on page 225.
Icon	File (`.gif` or `.bmp`) containing the icon graphic to appear with the label. (Optional)
Parameter	The parameters specified here are the parameters passed to the request. If there are no parameters to be used, leave this column empty.
Request Name	Name of the action or command that is triggered by the menu command. The Request can be user-defined, system-defined, or an external application. If the request name is for an external application, it is defined as the *`LogicalAppName`* in the application registration file (`/opt/OEMF/V6.0/GUIC/oemf/config/extappux.dat` for UNIX systems or `\Program Files\HPOVCA1.02\config\extapppc.dat` for the Windows NT operating system). For information on registering external applications, see the *HP OpenView Communications/Service Assurance Installation Guide*.

Column	Description
Shortcut	The keyboard shortcut that can be used to activate the corresponding menu command. Menu command shortcuts can be single letter keys or a combination of a letter and modifier keys, such as Ctrl+Shift+M. When the menu command is associated with a single letter key, the letter is case-insensitive. If the key is explicitly associated with the Shift key, the letter must be capitalized. Shortcuts may include regular alpha-numeric characters and the following keys: • **Shift** • **Ctrl** • **Enter** • **Esc** • **F1** to **F12** • **Home** • **End** • **Page Up** • **Page Down** • **Insert** • **Delete** • **Backspace**

Defining the Parameters

The parameters specified are the parameters passed to the request on the command line. Parameters are defined by Type and Value in the Parameters Dialog as shown in Figure 7-16. Parameters are specific to a particular request.

Figure 7-16 Parameters Dialog

Defining the Menu Positions

The `Position` column in the `Menubars` panel is used to define the position of the menu command within the menubar. Two methods are available for defining menubar position.

To define the position using letters:

Enter the one or two letter code for the position directly in the Position cell.

The first letter indicates the position of the menu in the menubar. The second letter indicates the position of the menu command under the menu indicated in the first digit. For example, AA indicates that this menu command is placed as the first menu command under the first menu on the menubar.

- A single letter indicates a new menu on the menubar. For example, the `File` menu is often located at position A, and the `Edit` menu is often located at position B.
- A two-letter code represents a submenu command. For example, AA often represents the `File:New` option, and BA might represent the `Edit:Cut` option. The first letter of the code indicates the location along the primary menubar; the second letter of the code indicates the sequence of menu commands under the primary menu.

NOTE Some commands on the menubar are predefined; others are user-defined. Although you can specify a location on the menubar, predefined menu commands have priority over menu locations and the user-defined menu command may be automatically moved. Menu locations are relative to existing menubar components.

To define the position using the Position Dialog:

1. Click in the `Position` cell, then click `[...]`.

   The Position Dialog appears as shown in Figure 7-17. This Position Dialog shows the positions of the default menu commands defined during the topology server system configuration procedure.

Figure 7-17 Position Dialog Box

2. Move the new menu command to the desired location by dragging or clicking `[Up]` and `[Down]` as necessary.
3. Click `[OK]` to complete the menubar entry.

   The Position Dialog shows the menu structure in tree-form.

Adding a New Menu Command

For example, if a trouble ticketing application is integrated with OV Topology Server, the following procedure adds a menu commands for Create/Update and Query under a new menu command, TroubleTicket:

1. At the top of the right frame, select ProblemPresenter from the Presenter Type list.
2. At the top of the right frame, select role from the Roles list.
3. Enter the following entries in the table of menu commands:

   Enter **TroubleTicket** for Label, then enter **B** for Position. Click [V] to verify this row and add an empty row.

   Enter **Create/Update** for Label, **BA** for Position, and **TroubleTicketApp** for RequestName. Click [V] to verify this row and add an empty row.

   Enter **Query** for Label, then enter **BB** for Position, then enter **TroubleTicketQueryApp** for RequestName. The RequestName is the logical name of the application.
4. Click [V], and click [Save].
5. Close and open the Problems Presenter. The menu command appears.

Configuring Save Session for Other Roles

The function Session:Set home session in the topology GUI presenters is accessible only to the network administrator (oemfadm), by default. However, it can be made available to other users through the Admin Panel. To enable this function for other users:

1. Log on as oemfadm to the topology server machine and invoke the Admin Panel.
2. Select the Menubar datatype from the left frame.
3. Select the role that will include Session:Set home session menu command from the Role list.
4. Select the presenter that should include the menu command from the Presenter list.
5. Enter the values shown in Table 7-11 in the Menubar table.

   Click [V] to validate each row before adding another row.

Table 7-11 Save Session MenuBar Settings

Column Name	Entry for `Session` menu command	Entry for `Session:Set home session`
`Label`	**`&sessionOperations`**	**`&saveAllConfig`**
`Position`	**`A`** (if other menu commands are configured, specify the next available location)	**`AA`**
`Parameter`		`Type:` **`string`** `Value:` **`OVCSASaveSessionCB`**
`Request Name`		**`CallScriptCallback`**

Leave the shortcut and icon fields empty.

6. Click `[Apply]` to apply the new menu command.
7. Click `[Save]` to save the entry.

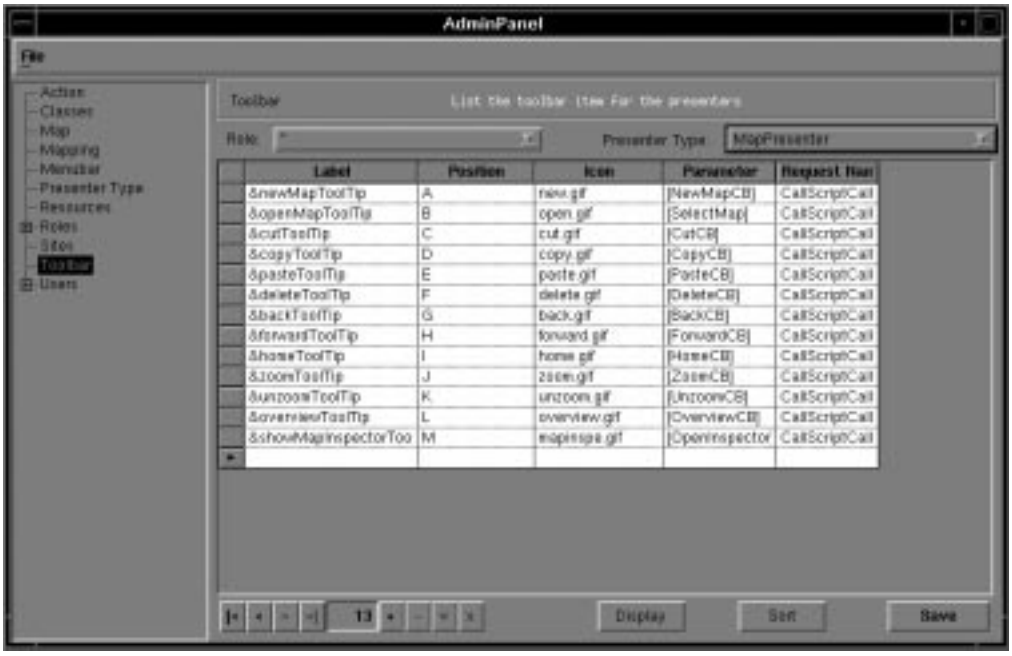
   To provide the Set home session function to all five presenters, repeat Steps 4 through 7 for each presenter.

The menu command `Session: Set home session` is then displayed in the configured presenter for the configured role.

Define Toolbar Buttons

Using the Toolbars datatype, you can define toolbar buttons and their commands, customized by role for the map presenter windows. All users view the same toolbar. The lists at the top of the panel are used to specify which roles will display the associated toolbar buttons. The toolbar is only available for the map presenter.

Figure 7-18 Toolbars Panel



During installation, the toolbar buttons are defined for the roles, * and admin.

To add new buttons to the toolbar, add a new row to the toolbar table. The following table describes the information to be specified for each toolbar button.

Column	Description
Label	The tool tip that appears when the mouse rests over the toolbar button, as it will be displayed in the presenter window or use a localizable string for string substitution. The label can be localized if you use an **`&string`** for the label field in the Admin Panel, then store the actual label in the `/opt/OEMF/V5.0/ilog/oemf/present/oemf_msg.dbm` message catalog.
Position	Used to define the position of the toolbar button. For information on setting position, see “Defining the Toolbar Button Positions” on page 231.
Icon	File (`.gif` or `.bmp`) containing the icon graphic to appear on the toolbar button.
Parameter	The parameters specified here are the parameters passed to the request. If there are no parameters to be used, leave this column empty, `Undefined`.
Request Name	Name of the action or command that is triggered by the toolbar button. The Request can be user-defined, system-defined, or an external application. If the request name is from an external application, it is defined as the *`LogicalAppName`* in the application registration file (`$GUIC/oemf/config/extappux.dat` for UNIX systems or `\Program Files\HPOVCA1.02\config\extapppc.dat` for the Windows NT operating system).

Defining the Parameters

The parameters specified are the parameters passed to the request on the command line. Parameters are defined by `Type` and `Value` in the Parameters Dialog as shown in Figure 7-19. Parameters are specific to a particular request.

Figure 7-19 Parameters Dialog

Defining the Toolbar Button Positions

The `Position` column in the `Toolbars` panel is used to define the position of the toolbar button within the toolbar. Two methods are available for defining toolbar button position.

To define the position using a letter:

Enter the one letter code for the position directly in the Position cell. This letter indicates the position of the button in the toolbar. For example, C indicates that this button is placed immediately to the right of the button with position B.

NOTE Some buttons on the toolbar are predefined; others are user-defined. Although you can specify a location on the toolbar, predefined toolbar buttons have priority over user-defined toolbar buttons. Toolbar button locations are relative to existing toolbar components.

To define the position using the Position Dialog:

1. Click in the `Position` cell, then click `[...]`.

   The Position Dialog shown in Figure 7-20 appears. This Position Dialog shows the positions of the default toolbar buttons defined during the GUI Server system configuration procedure.

Figure 7-20 Position Dialog Box

2. Move the new menu command to the desired location by dragging or clicking `[Up]` and `[Down]` as necessary.
3. Click `[OK]` to complete the toolbar button entry.

   The Position Dialog shows the toolbar structure in tree-form.

Set New Actions for Actions Menu

Actions are the commands accessible through the contextual menus in the presenter windows. Actions may be used to invoke presenter functions, third-party applications, or scripts. Actions can also be used to send requests to host applications. The only applications that should be launched via an Action are applications that do not require access control; other applications should be launched via a menu command.

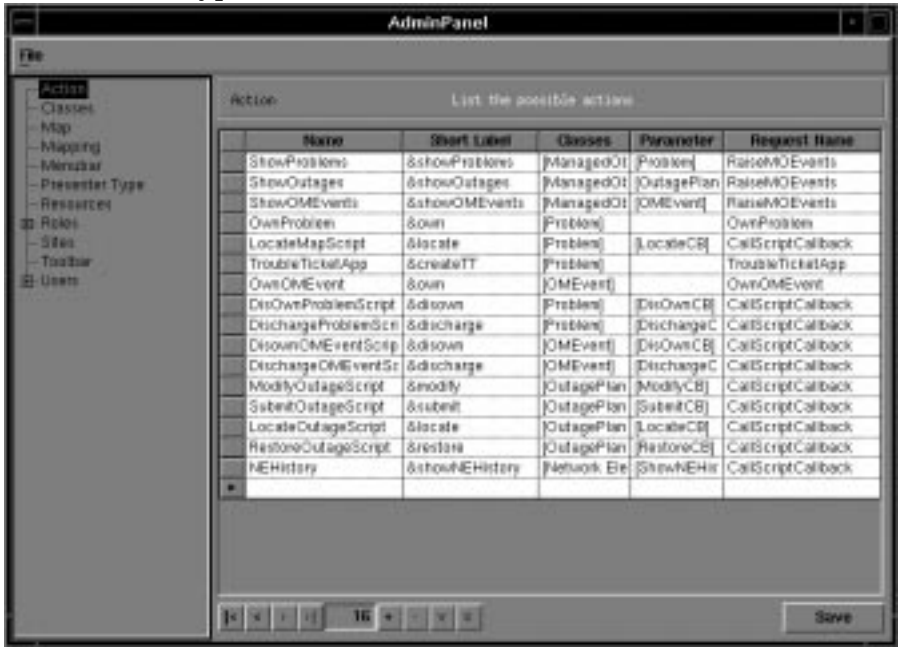
The benefits to launching applications from the Actions menu are:

- The application usage can be partitioned so, for example, the application is only accessible from a selected link.
- The application is available from the context-sensitive pop-up menu.

All actions accessible to users via the presenter windows must be defined using the `Action` function.

Figure 7-21 shows the Actions panel with the default set of actions defined for the topology server presenters:

Figure 7-21 Actions Datatype

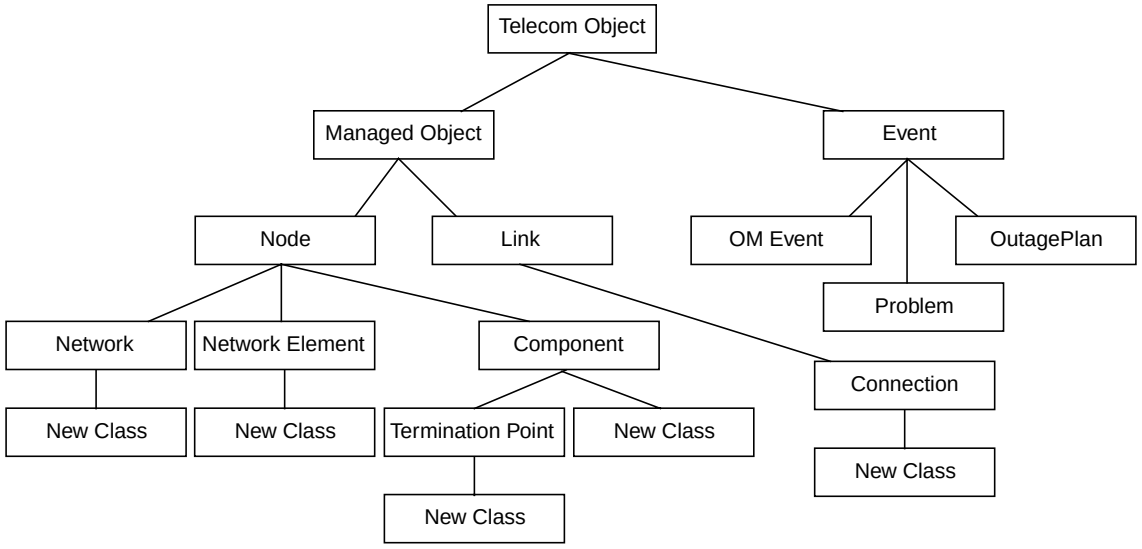


The following table describes each of the columns in the `Action` panel.

Column	Description
Name	Name of the action as referenced within the Admin Panel. This name has no value outside the Admin Panel.
Short Label	Name of the command as it should appear if the action refers to a contextual menu command or a tooltip. This label can also be a key to the message database file if prefixed with an ampersand (&). To localize this label, map the short label to the menu command in the `$GUISDIR/oemf/data/oemf/present/oemf.dbm` file.
Classes	Used to define the context in which the command can be triggered. You can define a set of dynamic classes for each command. Click in the Classes column to display the `[...]`. Click `[...]` to open the Parameters dialog box with two columns: `Type` and `Value`. In the Parameters dialog box, specify the classes or parameters for which this command should be available. For the `Type`, use `string`, and enter the name of the class for which this command should be available in the `Value` cell. Figure 7-22 on page 235 shows the available classes. Applying an action to a class also applies that action to all subclasses. For example, applying an action to the Node class associates that action with all objects of Network, Network Element, Component, or Termination Point and all user-defined classes under these classes. The network administrator defines new classes using the Telecom Configurator. For information on the Telecom Configurator, see the *HP OpenView Communications/Service Assurance Configuration Guide*. Click `[Apply]` to confirm the entries. The dialog box closes. This column will contain the word `[Defined]` or display a value to show that there are parameters defined for this action. If no classes are applicable, leave this dialog box empty.

Column	Description
Parameter	List of parameters that are transmitted to the request. See “Defining the Parameters” on page 224 for more information.
Request Name	Name of the action or command that is triggered by the menu command. The Request can be user-defined, system-defined, or an external application. If the request name is from an external application, it is defined as the *`LogicalAppName`* in the application registration file (`$GUIC/oemf/config/extappux.dat` for UNIX or `\Program Files\HPOVCA1.02\config\extapppc.dat` for Windows NT operating systems)

Figure 7-22 Available Classes



8 Other Administrative Utilities

This chapter details the utilities and programs to be run periodically to ensure that OV Topology Server is working effectively and efficiently. All utilities should be run by the `oemfadm` user on the topology server, unless otherwise stated.

Administering Event Correlation

The OV Topology Server utilities that establish logical connectivity and update event correlation are:

- `fmslogconadmin`—Establishes logical connectivity between devices for correlating alarms from objects not physically connected or sharing a containment relationship.
- `fmseccfgupd`—Updates the event correlation rules online.

The following sections explain these commands in detail.

Establishing Logical Connectivity

Event messages from unconnected objects can be linked and correlated within OV Topology Server. However, event correlation requires that the devices whose alarms are correlated share a containment relationship or that they be linked logically or physically. The `fmslogconadmin` command-line utility establishes logical connectivity between managed objects that are not physically connected but whose alarms must be correlated.

The command syntax is:

The command can be executed at any time and does not require the system to be shut down. This utility is not distributed. It must be run in the location in which the managed objects reside. The logical connectivity can be established only for objects in the same location. Cross-location logical connectivity is not possible.

Updating Correlation Rules

The `fmseccfgupd` command-line utility updates the event correlation rules on the FM Server without re-starting the server. The command syntax is:

`fmseccfgupd`

After running this utility the event correlation rules are updated.

This utility reads and updates the configuration per the Correlation Gateway File. It restarts the embedded ECS engine and reloads the circuit, fact store, and data files as applicable. If you are using FM Server OEMF-EC, this utility reads and updates the correlation configuration per the FM Server event correlation rules files.

The following entry in the `syslog` file indicates that the event correlation rules have been updated:

```
Information EVC_0001 EC configuration files loaded
successfully
```

To change the OEMF EC configuration you must:

1. Modify the configuration in the configuration files `ovfmpevcd.conf`, `ovfmpcgw.conf`, and `fmpecrules.conf` in the directory `$FMSETC/share/newconf` on the topology server.
2. Run `fmseccfgupd`.

9 Troubleshooting

This chapter details the utilities and programs to be executed, and files to be checked for problems that occur while using OV Topology Server. These cover:

- Verifying server status
- Checking the contents of the log files
- Understanding error messages listed in the syslog

While all HP OpenView products go through exhaustive testing prior to release, occasionally some defects or problems are discovered after the product is released. Check out the available HP OpenView software patches which solve some of the known product defects.

Verifying Server Status

You can verify the status of the topology server components any time using the command line utilities:

fmsstatus	to check status of the Fault Management (FM) Server processes. Refer to the section “Verifying the Status of FM Server” on page 33 for details.
guisstatus	to check the status of GUI Server processes. Refer to the section “Verifying the Status of GUI Server” on page 39 for details.
oemfstatus	to check the status of all OVC/Assurance processes. The following subsection describes the syntax of this utility.

Verifying the Status of the Topology Server

You can check the status of all the topology server components using a server status checking utility. To verify the status of all topology server processes, use the command:

`oemfstatus`

This utility displays the status of the all the topology server processes in the format of the individual server types (FM, and GUI Server). The order in which the status messages are displayed is:

- FM Server
-
- GUI Server

The status messages for each of the servers are the same as the messages displayed when the individual server utility is executed.

Understanding Errors Listed in the syslog

OV Topology Server writes all information and other messages into the `/var/adm/syslog/syslog.log` file. If you should encounter any problems while using OV Topology Server, check the syslog file for OV Topology Server messages. These messages are in the following format:

```
message_number    message
```

Messages belonging to OV Topology Server are prefixed either `ovfmp` or `fmpmd`.

The following is an example of an entry in the syslog file:

```
Feb 26 18:47:00 hpsgnco ovfmpevcd[2360] Information EVC_0003
EC starting in normal mode
```

If you need any explanation regarding the message listed, you can invoke the Error Checking utility that displays the details of the error messages logged in the syslog file.

To invoke the utility, enter:

`oemferr {`*`message_number`*` | list}`

The following information is displayed for the message:

- Message level (this can be Information, Error, Warning or Notice)
- Mnemonic of the module that sends the message
- Message
- Cause of the message
- Action - the remedial action, if any, to be taken
- Comment, if any

Follow the suggested course of action to correct the situation.

To view the entire list of error messages associated with the topology server, enter the command as follows on the topology server:

`oemferr list`

FM Server Does Not Startup After System Re-boot

If, after the FM Server machine is rebooted, the FM Server is unable to start up, it could be because OV has been automatically restarted when the system booted up.

In that case, stop OV using the command `ovstop`. Then edit the file `/etc/rc.config.d/ov500` to disable the OV auto-restart after reboot:

```
set START_OV500=0
```

Save the file and exit.

Now restart the FM Server using the `fmsstart` utility. The next time your FM Server machine reboots you should not have this problem.

Issues Related to Problem Presenter Display

The following problems relate to managed objects configuration.

Shortname *unknown* in Problem Presenter Display

New devices may be added to the network and linked to OV Topology Server, before their details are configured into the system. If alarms are emitted by these newly connected devices, then the column `NE ShortName`, in the problem presenter, will have `*unknown*` displayed under it. In addition, the network shortname, NE shortname might be `*unknown*`. Besides the status propagation behavior is not defined for the object or the its parent if the object does not exist.

After the topology database is updated, the existing problem continues to display the shortnames as `*unknown*`, even though new alarms arrive to update them.

To correct this error, discharge the corresponding problems. On receipt of the next alarm from these newly configured devices, a new problem record is created. Then this column (`NE ShortName`) displays the appropriate information.

Incorrect Information in Problem/Map Presenter

Information displayed in the problem presenter or map presenter is meaningless.

This problem can occur in the case of alarms sent via CMIP.

To correct this error, discharge the corresponding problems. Check to ensure that the NE are connected to the topology server, and reconfigure the object correctly to ensure that the topology server receives the alarm messages.

Alarm Sent to Non-Existing Object

When an alarm is sent to a non-existing object, the problem is created in the Problem Database. Error **PRS_0136** is logged to inform the user that the domain of this problem is set to `unknown_managed_object`. There is no associated Managed Object; hence, no known domain.

If `get_problem_data_by_filter` is invoked for all problems, **PRS_0107** is logged in the Problem Database.

To correct this error, the domain of the problem must be set to a known managed object.

Unlocking Sessions

By default, only the administration utilities lock the presentation database, permitting only one user read-write access at any one time. If a user exits one of these processes improperly, such as at a system crash, the username may still be bound to the session. When the user attempts to log on again, a message displays indicating the session is read-only.

The administration utilities are:

- `guidbadmin`
- `fmshmimport`
- `fmsopcfg`
- `oemflinkmap`

There are two ways to remove a session lock for a user's login.

- To remove the session lock from the Admin Panel:
  1. Click the `Users` datatype on the left frame.
  2. Click a username from the list of users in the left frame.
  3. Verify the username in the top box on the right frame.
  4. On the right panel, under the username, click `[Locked Session]` to unlock the session.
- To remove the session lock via the command line:

  Run `/opt/OEMF/V5.0/GUIDB/oemf/util/resetlock`.

Monitoring and Resolving Replication Conflict

The GUIDB is designed to eliminate most common replication conflicts. Some of the more infrequently used tables may still generate replication conflicts which are not automatically resolved. The sections below are instructions on what to do with replication problems on specific tables.

Replication conflicts may be monitored via the Oracle Replication Manager. For each connection the administrator should routinely check for replication conflicts. Replication conflicts are found in the replication manager under the Database `Connections->Connection->Administration->Local Errors`.

Once a replication conflict arises, it must be resolved before attempting the transactions again. Typically the conflicting rows are removed from one of the databases before attempting to execute those transactions again. Once the conflict is resolved the failed transaction will be either cancelled or re-attempted.

If the transaction was inserting the row into the database in which the offending conflict was cleared, then the user should re-attempt the transaction. The transaction may be selecting the specific Local Error and opening the properties window.

If the transaction was attempting to add a row to the database in which the offending conflict was not cleared then the replication error should be cleared.

Avoiding and Resolving Replication Conflicts in Map

When a replication conflict occurs in a map, each site will see the local version of the map until the replication conflict is eliminated. A map replication conflict is taken care of by deleting the map from one of the systems involved in the replication conflict. The map from the other system should be properly replicated during the next replication cycle.

To avoid replication conflicts with user maps, a naming convention may be used. To avoid replication conflicts for Server Maps, administration rules can be used. For example: administrators would only be responsible for their partition of the network. Therefore they should not be modifying another partition.

Using the Admin Panel to Avoid and Resolve Replication Conflict

Only the Admin Panel modifies a mapping table. Replication conflicts can be avoided by running this tool at a single site. The administrative tool can also be used to resolve the replication conflict. Start the tool on one of the servers involved in the replication conflict. Remove the entry associated with the replication conflict. `At Mapping` should be replaced at the next replication cycle.

Avoiding and Resolving Replication Conflicts in User Session

Replication conflicts in the `UserSession` arises when a user saves a session from more than one site at the same time. To avoid these replication conflicts, each user should be allocated their own login. They should also not be logged into the different sites at the same time.

To resolve the replication conflict, run the following sql commands from one of the systems involved in the replication conflict:

```
sqlplus guidbadm/<password>@guidb
SQL> delete from parameter where paramid = select paramid from
     usersession where

     username='USER' and rolename='ROLE' and presenterid='N';

SQL> delete from usersession where
     username='USER' and rolename='ROLE' and
     presenterid='N';
```

Where USER, ROLE and N are from the local error generated by the replication conflict.

Troubleshooting CORBA Problems

Following are some recommended practices that can help you prevent, isolate, and recover from CORBA problems.

- Check the information logged in `/var/adm/syslog/syslog.log`.
- If you received an error message, use the `/opt/OVCORBA/bin/ovlerch` command to list the chain of related errors. This will help significantly in tracking down the specific cause of the error message. See the online Reference Pages manual for more details about the `ovlerch` command.
- Check for stack traces and log files in `/var/tmp/<service_name>/<service_name>.core` and `/var/opt/OVNSLS/notifserver/notifserver.core`. These traces are generally small and fairly useful.
- Ensure that the system meets the recommendations discussed in the *Installation Guide*, which includes hardware, software, and configuration prerequisites.
- Use message logs to help isolate problems. For more details, see “Searching and Managing Logs” on page 253.
- Make backup copies of files that you are modifying. The original files provide a way of restoring a known operational configuration. If you can correct a problem by reverting to the original files, you can isolate the problem to the changes you made to these files.
- See Table 0-1 on page HIDDEN for additional assistance on specific problems and errors.
- To check whether IORs are valid and servers are operational, use the one of the commands illustrated in these examples:

  `/opt/OVCORBA/bin/ovcorba_admin -status`

  `/opt/orbplus/sbin/orb_admin ping < /etc/opt/OVNSLS/IORs/NotificationService`

  `/opt/orbplus/sbin/orb_admin display < /etc/opt/OVCORBA/IORs/TransactionService`

CORBA Problems / Solutions

Table 0-1 lists possible problems that you might encounter when configuring HP OVC CORBA along with suggested solutions.

Searching and Managing Logs

CORBA's Message Logging Service is used by OV Topology Server collection managers to log error, warning, and informational messages. This service as well as the `syslog` and the specific module logfiles together provide valuable information for troubleshooting.

Searching HP OVC CORBA Log Messages

Table 9-1 describes the command to use for displaying message logs (resides in `/opt/OVCORBA/bin`).

Table 9-1 Message Log Searching Command-Line Tools

Command-line Tool	Function
`ovdumplog`	Displays log entries from a CORBA log within an installation; allows selection of messages to dump based on date, time, origin, identifier, host where logged, and/or message text. Ranges of values not supported.
`ovlognotifs`	Writes to `stdout` all messages logged and message logging alarms generated while `ovlognotifs` is running; entries are continually updated as new messages arrive.

Ovdumplog

A filter is specified as a string using the `-f` option of `ovdumplog`. The format of the filter is a UNIX regular expression. You may need to enclose the filter string in double quotes to prevent the shell from interpreting special characters such as *, [, and \.

The filter is used to restrict the set of messages dumped to standard output. It is applied to each message when it is formatted for output. If the filter matches the formatted message, then the formatted message is output.

As an example, to display the contents of the CORBA error log, execute:

```
/opt/OVCORBA/bin/ovdumplog
```

The next example shows how you would restrict the messages extracted from the error log file in CORBA to those containing a message text that includes the word “My”, separated from the word “service” by zero or more characters:

```
/opt/OVCORBA/bin/ovdumplog -f "My*service"
```

The following example dumps MOI Handler errors found in the CORBA Error Log:

```
/opt/OVCORBA/bin/ovdumplog -f OEMFMOIHandler
```

Similarly, a filter specifying `OEMFManagedObject` can be used to find Managed Object management errors, `OEMFOutagePlan` for outage plan management errors, `OEMFProblem` for problem management errors, etc.

For example, to dump messages in the CORBA Error Log relating to topology server components, use the following:

```
/opt/OVCORBA/bin/ovdumplog -f "OEMF"
```

```
/opt/OVCORBA/bin/ovdumplog -f "fms"
```

Ovlognotifs

Following are some examples of how `ovlognotifs` can be used.

- To view messages being logged on the current host, type:

  ```
  ovlognotifs
  ```

- To view detailed messages logged on hosts `saturn` and `jupiter` (names may need to be qualified if in different domains), type:

  ```
  ovlognotifs -d -h saturn -h jupiter
  ```

- To view messages with full detail in `ovdumplog` format, type:

  ```
  ovlognotifs -D
  ```

Searching Other Error Logs

Syslog

- Note that ORBPlus errors are logged to:

  `/var/adm/syslog/syslog.log` for HP-UX,

  or to:

  `/var/adm/messages/syslog.log` for Solaris.
- Many of the HP OVC CORBA Services will also write to `syslog`.

CORBA Service Logs

Other CORBA Service logs include:

`/var/tmp/tp_server/*`

`/var/tmp/ams_server/*`

`/var/tmp/unified_server/*`

`/var/tmp/msglog_server/*`

Notification Service Logs

The Notification Service logs error information in the following files:

`/var/opt/OVNSLS/notifserver/emergency.err`

`/var/opt/OVNSLS/notifserver/notifserver.log`

Managing HP OVC CORBA Message Logs

There are some tools available in `/opt/OVCORBA/bin` to help you manage your message and error logs. These are listed in Table 9-2.

Table 9-2 Message Log Management Command-Line Tools

Command-line Tool	Function
`ovlogalarm`	Displays and sets the alarm threshold for a log.
`ovlogrollover`	Displays and sets the rollover behavior for a log.

Table 9-2 Message Log Management Command-Line Tools

Command-line Tool	Function
`ovlogsev`	Displays and sets the severity for a log message type in a particular log.
`ovlogsize`	Displays and sets the maximum logsize for a particular log.
`ovlslogs`	Lists the names of the current logs available in an installation.
`ovlstypes`	Lists the log message types registered with a particular log.

For more details about these commands, see the online Reference Pages manual available with the online documentation.

Troubleshooting Topology GUI Related Problems

This section provides solutions to the errors that could occur while running the GUI Server and the topology GUI presenters.

Problem Presenter Not Being Updated

You have a problem condition in the problem presenter which is in an owned state. When you attempt to discharge it, a dialog box pops up full of OVError type messages indicating that this problem condition does not exist in the topology server.

Close the problem presenter and open it again. The problem should not appear in the new presenter window. Check `ShowAll` to ensure that the problem is still not present.

However, if the problem still persists, close all problem presenters and log on again. This will force the GUI Server to reload all information from the FM Server, and thus update the problem status.

Setting the Background Map for the Map Presenter

There is an option in the mouse menu in the map presenter to set background map using the `Set map...` function. Should this fail, you can use one of the following two options:

- The background map for the map presenters must be set using the `Admin Panel` as described in section.
- Alternatively, you can set the background map through the import file when creating the object using the MOI Handler. For more details, refer to the section “Format of the Intermediate File” on page 120.

NOTE Background images are not updated on other topology GUI sessions until the operator logs on to a new session.

Alt Key Does Not Work

On some HP-UX platforms, the `mod1` modifier map is set to `Meta_R (0xa), Meta_L (0xb), Mode_switch (0x36)`.

To correct the behavior:

1. Remove `Mode_switch` from the mapping:

   `/usr/bin/X11/xmodmap -e "remove mod1=Mode_switch"`

2. Restart the topology GUI.

Using Trace Logs

All utilities and applications write summary log information in the syslog file (`/var/adm/syslog/syslog.log`). Server modules write messages in individual log files under the server's log directories. Besides these messages, there are other messages that can be logged by the components, utilities, and application to help diagnose processing errors or status. These messages are not ordinarily logged; however, their logging can be enabled using the trace environment variables. This section describes how these environment variables are set for each of the servers.

There are trace levels for various modules that can be set to assist in troubleshooting OV Topology Server.

Note that these trace logs must generally be enabled only at the request of the HP support engineer as these logs do not have a maximum size and could potentially use up much of the disk space leading to performance issues, or even "disk-full" induced operational failure.

Using Trace Logs on the FM Server

On the FM Server the trace logs can be found under the directory `$FMSVAR/share/log/trace`. On the FM Server, trace can be activated/deactivated for the following:

- NSM module of the FM Server
- Collection Managers

To activate the trace log:

Step 1. Log on as user `root` on the FM Server.

Step 2. Shut down the FM Server using `fmsstop`.

Step 3. Shut down OpenView processes using `ovstop`.

Step 4. Set the environment variable to activate the trace log. See below for environment variable descriptions.

Step 5. Start the FM Server process by executing `fmsstart`. This starts up OpenView processes and then the FM Server modules.

The environment variables for the FM Server are:

`MIS_TRACE_LEVEL`

Is used on the FM Server to set trace for the NSM module. Setting this environment variable logs NSM process messages into the file `ovfmpnsmd.trace` in the trace directory.

The valid values to which this environment variable can be set and their trace level are listed below:

0	=None. No trace messages will be logged when the environment variable is set to 0.
1	=Any event messages. Set to this value *all trace messages* that are generated by NSM will be logged in the trace file.
2	=Error trace. All *error messages* for NSM will be logged in the trace file.
3	=Logic flow trace. All *messages related to the logical flow* of the process will be logged in the trace file.
4	=Data trace. All *messages related to the data flow* will be logged in the trace file.
5	=Info trace. All *informational*

messages will be logged in the trace file.

TOP_TRACE_LEVEL

Is used on the FM Server to set trace for the topology database APIs only. Setting this environment variable logs topology database API run process messages into the file `ovfmpnsmd.trace` in the trace directory.

The valid values and their trace level are the same as for the **MIS_TRACE_LEVEL**.

Generally, it is more beneficial to run both **MIS_TRACE_LEVEL** and **TOP_TRACE_LEVEL**.

OVTRACE_LEVELS

Is used on the FM Server to log collection manager messages. When this environment variable is set, all collection managers will log messages under the trace directory on the FM Server. The trace file name will be as follows:

servername`.trace.`*data-timestamp*

The *date-timestamp* is of the format *yyyymmddhhmiss*.

This environment variable must be set to **1-200.30** to enable tracing.

To deactivate the trace log:

Step 1. Log on as root on the FM Server.

Step 2. Shut down the FM Server using `fmsstop`.

Step 3. Shut down OpenView processes using `ovstop`.

Step 4. Unset the environment variable to deactivate the trace log.

Step 5. Start up the FM Server by executing `fmsstart`. This starts up the OpenView processes and then starts up the FM Server modules.

Using Trace Logs on the GUI Server

For the GUI Server the trace level is set in the GUI Server environment variable file, `/opt/OEMF/V5.0/GUIS/oemf/util/guisenv`. The environment variable that is used to activate/deactivate trace log is `iltLOGLEVEL`. On the GUI Server, the trace logs can be found under the directory `$GUISVAR/share/`*`hostname`*. Under this directory the messages are logged separated by the application logging them and filed under timestamped log files.

To activate or deactivate the tracing and logging on the GUI Server:

Step 1. Log in as `root` on the GUI Server.

Step 2. Shut down the GUI Server using `guisstop`.

Step 3. Edit the environment variable file to set the trace level.

```
ILTLOGLEVEL=level
export ILTLOGLEVEL
```

The valid values of *`level`* are described below.

NOTHING	This implies that no trace messages will be logged. This is the default value of the environment variable.
ALL	All trace messages will be logged in the log file.

Step 4. Start up the GUI Server by executing `guisstart`.

Using Trace Logs on the Topology GUI

For the topology GUI, the trace level is set directly in the script that starts the topology GUI process. The command-line argument that is used to activate/deactivate trace log is `iltLogLevel`. On the topology GUI machine, the trace logs can be found under the directory:

- Windows NT operating system: `\Program Files\HPOVCA1.02\bin\log\`*`hostname`*`\Presenter_PresenterApplication\`
- Unix systems: `/var/opt/OEMF/V5.0/GUIC/share/log/`*`hostname`*`/Presenter_PresenterApplication/`

Under this directory the messages are in timestamped log files.

To activate the tracing and logging of the topology GUI process on the Windows NT operating system, enter the command:

```
present.exe -iltLogLevel ALL
```

NOTE An easier way to activate tracing and logging for this and future sessions is to move the `REM` statement in the `guistart.bat` file.

To activate or deactivate the tracing and logging of the topology GUI process on a UNIX system:

Step 1. Log on as `root` on the topology GUI machine.

Step 2. Edit the `/opt/OEMF/V5.0/GUIC/oemf/util/guicstart` file to set the trace level.

Edit the line containing `set iltLogLevel="-iltLogLevel` *`level`*", to set *`level`* to the desired value.

The valid values of *level* are:

NOTHING	This implies that no trace messages will be logged. This is the default value of the environment variable.
ALL	All trace messages will be logged in the log file.

Step 3. Start the topology GUI.

Troubleshoot Topology Database Inconsistencies

Run `fmstopofix` as `oemfadm` on the server on which the topology database has become inconsistent. It takes one argument, `logfile`.

If you do not specify the input file on the command line, the fix utility retrieves all inconsistencies logged for the current location and fixes all local inconsistencies first. Then, it writes all remote inconsistencies in log files, one file per location. After the inconsistencies are fixed, all inconsistencies records are removed from inconsistency log database. The log files are copied to the peer locations involved so the utility can use them to repair the inconsistent objects in that location.

If you specify the fix input file from the command line, the fix utility takes this input file to fix the inconsistencies contained in it. It may generate other log files as needed to fix peer location inconsistencies.

All results are logged to the `$FMSVAR/topofix/fmstopofix.report` file. Check this file for any problems that the `fmstopofix` utility could not handle. If there are any such inconsistencies, fix them manually to maintain the topology database consistency.

Syntax

`fmstopofix` *`logfile`*

Where:

`logfile` This is an optional parameter. If it is not specified, then the utility retrieves all inconsistencies logged for the current location and fixes all local inconsistencies first. Then, it writes all remote inconsistencies in their respective log files. The generated inconsistency log files are located in the `$FMSVAR/topofix/` directory.

If a file name is specified (with UNIX path, as the administrator must copy it from the source location), the utility takes this input file to fix the inconsistencies contained in it. It may generate other `topdb-fix.`*`location`* files that are needed to fix peer location inconsistencies. The generated inconsistency log files generated are located in the

`$FMSVAR/topofix/` directory.

This command may generate other inconsistency log files if it detects any inconsistency from the local topology database.

A Error Messages

This chapter provides a list of messages you may encounter while using OV Topology Server and the corrective action, if any, that you should take to overcome the error.

The messages explained in this chapter may be encountered while using command line programs (such as, `fmsstatus`). All message are listed in alphabetic order.

Command Line Programs

Message: ERROR: Cannot not copy *filename* from *source_directory* to *destination_directory*.

Cause: The specified file either does not exist or you do not have the correct access rights.

Action: Check for the presence of the file in the given source directory and check the access rights. If required, install the product being configured and begin the system configuration process again.

Message: ERROR: Cannot open or read file *filename*.

Cause: Either the specified file is missing or you do not have the correct permissions to access it.

Action: Use the UNIX `swinstall` utility to install the product on the machine again and then re-configure the product.

Message: ERROR: Environment time variable TZ has not been set!

Cause: The environment variable TZ which is required has not been set.

Action: Use the UNIX `export` to set and export the TZ variable.

Message: ERROR: FMPCFG for FMS has already been installed on this system.

Cause: You are attempting to install FMPCFG on a machine that already has it installed.

Action: If you wish to re-install FMPCFG, use the `swremove` utility to remove the installed FMPCFG, then install it again.

Message: ERROR: FMS is still running. Please shutdown FMS before proceeding further.

Cause: You have attempted to install an OV Topology Server product while the FM Server is running.

Action: Run `fmsstop` to shut down the FM Server, then run the utility again.

Message: ERROR: Oracle has returned the error displayed below. Refer to the Oracle7 Server Messages and Codes Manual for a complete description of the error returned.
<Oracle error message>

Cause: Oracle has returned the error displayed.

Action: Refer to the *Oracle7 Server Messages and Codes* Manual for a complete description of the error message displayed. Take the prescribed corrective action and then re-install FM Server.

If you persistently see the ERROR ORA-01555, contact HP Customer Support.

Message: ERROR: Oracle is not running. Please startup Oracle first before proceeding further.

Cause: Oracle should be started up before you can install the FM Server. You have tried to use the `fmssysconfig` utility without starting up Oracle.

Action: Startup Oracle and then configure FM Server.

Message: ERROR: *ovaddobj filename* returned an error code. Check the log file *filename* to determine the cause of the error.

Cause: The process for registering processes with OV DM TMN has returned an error.

Action: Check the specified file for the error message and take the required corrective action.

Message: ERROR: The DISPLAY environment variable has not been set.

Cause: You have tried to start a Presentation Client session on a HP-UX workstation without setting the DISPLAY environment variable.

Action: Use the UNIX `export` command to set and export the display environment variable, DISPLAY, then start the Presentation Client session.

`Message: ERROR: The fileset fileset chosen has not been loaded yet!`

Cause: You have tried to install a fileset that has not yet been loaded on the machine.

Action: Use the UNIX `swinstall` utility to load the required product on the machine and then configure it.

`Message: ERROR: The fileset fileset has already been installed on this machine.`

Cause: You are attempting to install a fileset that has already been installed.

Action: None.

`Message: ERROR: This script can be run by super-user only.`

Cause: You have tried to run an OV Topology Server utility that can be run only by the super-user (`root`).

Action: Log on as `root` and then invoke the utility again, or contact the system administrator to execute the required utility.

`Message: Information : FMPC is saving new configuration now!`

Cause: The configurator is saving modifications to the configuration files. You cannot start up the system during this process.

Action: Wait awhile and then run the startup utility again.

`Message: FMS is running, please shut it down before using fmssysconfig!`

Cause: You cannot update your installation while the FM Server is running.

Action: Run `fmsstop` to shut down the FM Server, and then run `fmssysconfig` again.

`Message: FMS is running, please shut it down before using fmsremove!`

Cause: You cannot run the `fmsremove` utility while the FM Server is running.

Action: Run `fmsstop` to shut down the FM Server, and then run `fmsremove` again.

`Message: FMS startup failed, check syslog for details!`

Cause: There has been an error while starting up the FM Server.

Action: Check the syslog for the error message and take appropriate action.

`Message: Oracle is not running, please startup Oracle before installation!`

Cause: The installation utility has detected that Oracle is not running. Oracle should be running before you can configure FM Server.

Action: Start up Oracle and then run the `fmssysconfig` utility again.

`Message: Install FM database encountered errors, see` *`filename`* `for errors.`

Cause: The system configuration utility has encountered some problems while trying to create the OV Topology Server database table.

Action: Check the FMPDBLOG for errors and take appropriate action.

Message: FMS not updated properly!

Cause: While running `fmsstart`, the system was not able to updated the FM Server databases properly.

Action: Run startup in recovery mode.

Message: Must be run by super-user.

Cause: The utility that you have tried to invoke can only be run by a super-user.

Action: Ask the System Administrator to login as `root` and run the required utilities.

Message: *port_number* port is already used by others.

Cause: The specified port number is reserved for OV Topology Server, and cannot be used by others.

Action: None

Message: *Product* has not been updated!

Cause: The specified product files have not been updated.

Action: Re-install the specified product.

Message: *Product* not installed!

Cause: This is an information message only. The specified product has not been installed or configured.

Action: Install and configure the product and then run the command again.

Message: *Program* cannot execute!

Cause: The system is unable to execute the specified program

Action: Re-install the program.

Message: *Program* returns unknown status!

Cause: The FM Server is in unknown status.

Action: Stop OV. Stop FM and then restart the programs.

`Message: Recovery startup failed, check syslog for details!`

Cause: The system was unable to startup in recovery mode.

Action: Check the syslog for the error message and take appropriate action.

`Message: User cannot execute!`

Cause: You have tried to run a utility for which you do not have appropriate access rights.

Action: Contact the system administrator.

`Message: Unable to add user` *`fmpadm`* `to Oracle, check` *`filename`* `for error.`

Cause: Unable to add the `fmpadm` user to Oracle.

Action: Check the Oracle error log file for details.

B OV Topology Server Problem Database Schema

OV Topology Server Problem Database Schema

This chapter provides the details of the problem database schema used by OV Topology Server which can be accessed users to extract information.

OV Topology Server Tables

OV Topology Server logs its alarm details in the following database tables which can be accessed by user applications, if required:

Table Name	Contents
alarm	Alarm details
alarm_additional_info	Additional information on alarms
ec_state	Event correlation details
problem_audittrail	Audit trail of activities on problem conditions
problem_state	Activity list on the state of problem conditions
problem_to_event_map	Log of events in the problem condition
alarm_additional_text	Additional text on alarms
problem_annotation	Annotation text
alarm_state_change	Store alarm status change

Table - alarm

This table is used to store the details of the alarm messages received, such as the particulars of the device emitting the alarm and cause and severity of the alarm. This 28 column file is indexed on the `alarm id`.

Table - alarm_additional_info

This table contains the additional information on the alarm. Only one record can exist for each alarm ID. A column `rec_num` tracks the number of records for the same alarm ID. These records together make up the text of the additional information for the alarm.

It is linked to the **Table alarm** by the alarm ID.

Table - problem_to_event_map

This table tracks the various events in the history of the problem condition. It consists of the problem condition ID and the event ID.

It links to the **Table alarm** by the `alarm id`.

Table - problem_state

This table records the various operations executed on the problem condition. For example, it records the time at which the problem condition was owned, the time of creation of the trouble ticket and related details.

This table is linked to the Tables **problem_to_event_map** and **problem_audittrail** by the `problem id`.

Table - ec_state

This table is used to store correlation information on the alarms. It lists the main alarm, such as the root cause alarm, and lists the related alarms or transient alarms correlated with it.

This is linked to the **Table alarm** by the `alarm id` column.

Table - problem_audittrail

This table maintains an audit trail of the various events in the life of a problem condition.

It is linked to the **Table problem_state** by the `problem_id`.

Table - alarm_additional_text

This table is used to store the additional text information for the alarms. Only one record can be stored for each alarm ID.

This is linked to the **Table alarm** by the `alarm id` column.

Table - problem_annotation

This table is used to store the problem annotation entered by the user.

This is linked to the **Table problem_state** by the `problem id` column.

Table - alarm_state_change

This table is used to store alarm status change.

This is linked to the **Table alarm** by the `alarm id` column.

The diagram in Figure B-1 shows the relation between these files. The details of the tables are listed in the following sections.

Figure B-1 Link Between the Tables

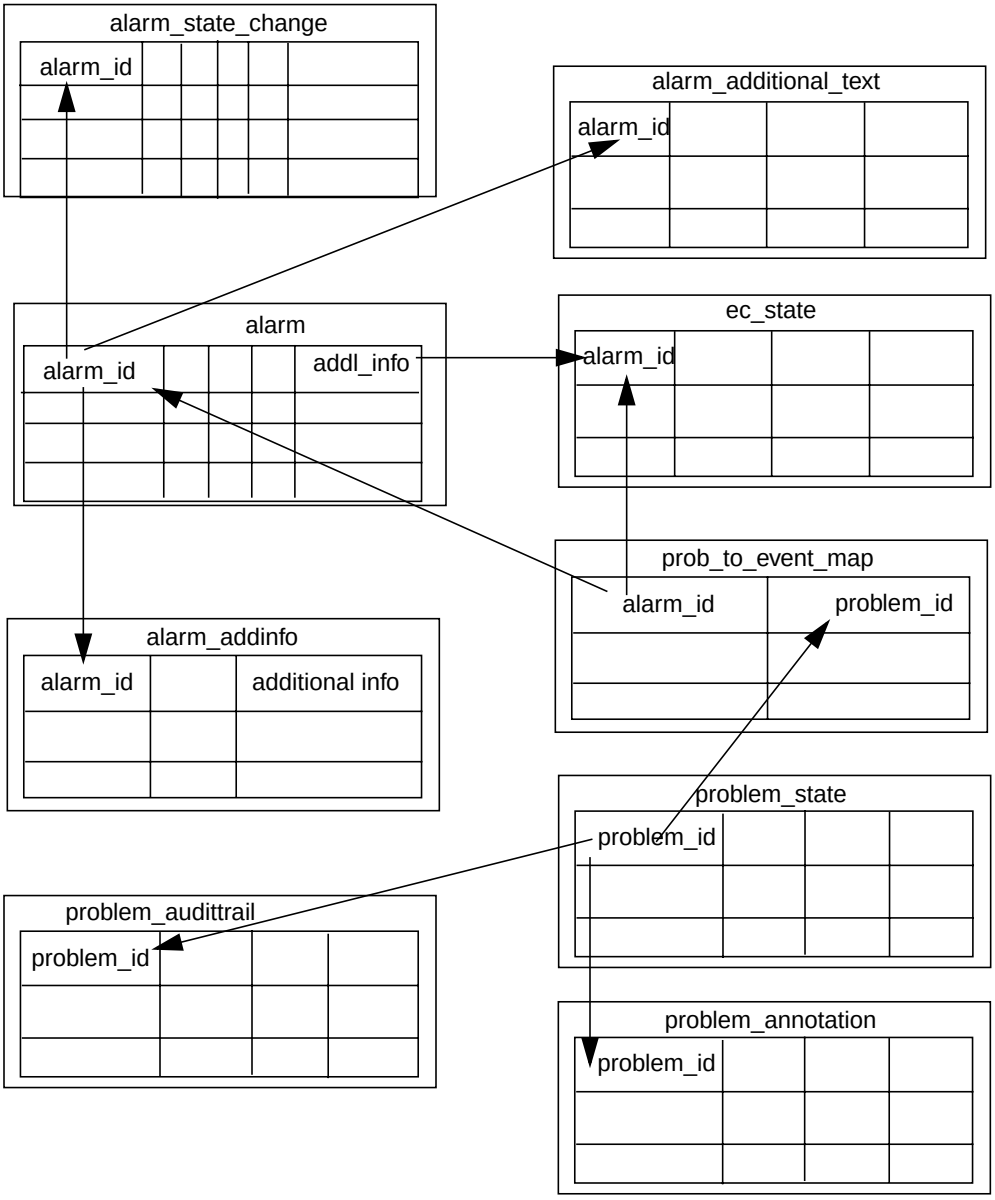


Table - alarm

Column Name	Type	Length	Nulls	Defaults
`alarm_id` Alarm ID as assigned by product	integer	38	no	no
`event_type` The alarm type	integer	1	no	no
`event_time` Time at which the event was first registered	date		no	no
`log_time` Time at which this record was inserted into this table	date		yes	no
`managed_object_class` The MOI to which the NE belongs	vchar	32	no	no
`managed_object_instance` The MOI to which the NE belongs	vchar	1024	no	no
`short_name` The short name of the mo	vchar	129	no	no
`network_moc` The network class to which the MOC belongs	vchar	32	no	no
`network_short_name` The short name of the network that contains the MOC	vchar	64	no	no
`network_equip_moc` The NE class to which the MOC belongs	vchar	32	no	no

Table - alarm

Column Name	Type	Length	Nulls	Defaults
network_equip_short_name The short name of the NE class which contains the object in alarm state	vchar	64	no	no
mo_id Object ID of the mo that emitted this alarm	vchar	256	no	no
mo_prop_alarm_status The propagated alarm status of the mo	integer	1	no	no
mo_admin_state The administrative state of the object	integer	1	no	no
notification_identifier	integer	38	no	no
probable_cause The probable cause of the alarm	integer	5	no	no
specific_problems The specific problem of the device	vchar	64	yes	no
perceived_severity The severity of the alarm	integer	1	no	no
trend_indication	integer	1	no	no
backed_up_status	integer	1	no	no
back_up_object	vchar	1024	no	no
threshold_info	vchar	96	no	no
monitored_attributes	vchar	256	no	no
proposed_repair_actions	vchar	352	no	no
correlated_notifications	vchar	2000	no	no
state_change_definitions_ind	integer	1	no	no

Column Name	Type	Length	Nulls	Defaults
additional_info_ind 1 to indicate additional_information is available; 0 indicate that it is not	integer	1	yes	no
additional_text_ind 1 to indicate additional_text is available; 0 indicate that it is not	integer	1	yes	no

Indices

Index	NameKeyed On
alarm_idx	alarm_id

Table - alarm_additional_info

Column Name	Type	Length	Nulls	Defaults
alarm_id Alarm ID as assigned by product	integer	38	no	no
additional_info Record number and the additional text of the alarm	long raw	2G	no	no

Indices

Index	NameKeyed On
additional_info_alarm_idx	alarm_id

Table - problem_to_event_map

Column Name	Type	Length	Nulls	Defaults
`problem_id` Problem ID assigned by product	integer	38	no	no
`alarm_id` Alarm ID as assigned by product	integer	38	no	no

Indices

None

Table - problem_state

Column Name	Type	Length	Nulls	Defaults
problem_id Problem ID as assigned by product	integer	38	no	no
log_time Time at which this record was inserted into this table	date		no	no
last_update_time Time at which the last update was made	date	4	yes	no
discharged_time Time at which the problem condition was discharged	date	4	yes	no
owned_time Time at which the problem condition was owned	date	4	yes	no
owned_by Id of the operator who owns the problem condition	vchar	256	yes	no
problem_hash_number	integer	9	no	no
cur_alarm_id Alarm ID of the current update alarm	integer	38	no	no
state State taken from the prob_audittrail where problem_id=pcid	integer	1[a]	yes	no
old_severity perceived_severity of the second largest alarm_id of the problem_to_event_map	integer	1	yes	no

Column Name	Type	Length	Nulls	Defaults
`cum_related_count` Cumulated related alarm count	integer	5	yes	no
`cum_transient_count` Cumulated count of transient alarms	integer	5	yes	no
`cor_state` The correlated state of the alarm	varchar	32	yes	no
`tt_created_time` Time at which the trouble ticket was created	date	4	yes	no
`tt_created_by` Id of the operator who created the trouble ticket	varchar	256	yes	no
`tt_id` Trouble ticket ID of the problem condition	varchar	32	yes	no
`annotation_ind`	integer	1	yes	0

a. Its value can is dependent on the eventid:

state for eventid
0 for PC_Owned
1 for PC_Unowned or PC_Created
2 for PC_Discharged
3 for PC_TT_Created or PC_TT_Updated

Indices

Index	NameKeyed On
prob_state_idx	problem_id

Table - ec_state

Column Name	Type	Length	Nulls	Defaults
`alarm_id` Alarm ID as assigned by product	integer	38	no	no
`majoralarmid` Alarm ID of the related/transient alarm	integer	4	yes	no
`ecstate`	integer	1	no	no
`related_count`	integer	5	yes	no
`transient_count`	integer	5	yes	no

Indices

Index	NameKeyed On
ec_state_idx1	majoralarmid
ec_state_idx2	alarm_id

Table - problem_audittrail

Column Name	Type	Length	Nulls	Defaults
`problem_id` Problem condition ID as assigned by product	integer	38	no	no
`audit_time` Time at which the audit entry was made	date	4	no	no
`action` Event type of the operation executed by the operator	integer[a]	2[b]	no	no
actioned_by Name of operator that performed the action. Required only for own, disown, discharge and tt_creations	varchar	256	yes	no
`tt_id` Trouble ticket ID	varchar	32	yes	no

a. This is the enum returned by the application when the event occurred. Valid values are:

0 = manage a problem
1 = own a problem
2 = Disown a problem
3 = Discharge a problem
4 = Create TT

b. ** If the operator exits without relinquishing ownership of the problem conditions and does not log in again before the timeout period expires then in the additionalinfo column the operator's name will be followed by "(*timeout*)". This indicates that the ownership had been disassociated by the system after the timeout period expired.

Indices

None

Table - alarm_state_change

Column Name	Type	Length	Nulls	Defaults
alarm_id Alarm ID as assigned by product	integer	38	no	no
state_change_definitions	long raw	2G	yes	no

Indices

Index	NameKeyed On
state_change_alarm_idx	alarm_id

Table - alarm_additional_text

Column Name	Type	Length	Nulls	Defaults
alarm_id Alarm ID as assigned by product	integer	38	no	no
additional_text	long raw	2G	yes	no

Indices

Index	NameKeyed On
additional_text_alarm_idx	alarm_id

Table - problem_annotation

Column Name	Type	Length	Nulls	Defaults
`problem_id` Problem condition ID assigned by product	integer	38	no	no
`annotate_time` Time at which the problem was annotated	date		no	no
`annotate_by` Name of operator who annotated the problem	vchar	256	no	no
`note` Annotate text	vchar	2000	yes	no

Indices

None

C Applications and Tasks

This appendix lists the applications and their associated task names that are provided with OV Topology Server.

List of Applications and Tasks

The following are the default applications and their related tasks that are provided with OV Topology Server. These application and task names will be displayed in the Operation Profile Configurator when the Application Maintenance function is invoked.

There are four applications:

- managed_object_application

  This is the application name associated with the actions taken with respect to managed objects via the managed object manager. This application is referred to as the map presenter.

- problem_application

  This is the application name associated with actions taken with respect to problems for managed objects via the problem manager. This application is referred to as the problems presenter.

- outage_plan_application

  This is the application name associated with actions taken with respect to outage plans for managed objects via the outage plan manager. This application is referred to as the outage plan presenter.

- om_event_application

  This is the application name associated with actions taken with respect to OM events for managed objects via the OM Event manager. This application is referred to as the OM event presenter.

The following four tables list the tasks associated with each of the applications mentioned above.

Table C-1 Tasks Associated with managed_object_application

Task Name	Description
get	This task must to assigned to all users being provided access to this application. It enables the user to view the Map Presenter. Enables the user to receive notifications from the FM Server. It also enables the user to query managed object information from the FM Server.
create	Enables the user to create managed objects.
delete	Enables the user to remove or delete managed object.
add_to_domains	Having created a managed object, this task enables the user to dynamically add this newly created object instance to a domain. Though this task can be selected and assigned to any operator, it will be made available only to the administrator in the Map Presenter.
remove_from_domains	When a managed object is removed from the FMS, it needs to removed from the domain, too. This task enables the user to remove the object instance from the domain.
clear_propagated_alarm_status[a]	Enables the user to clear the alarm status tagged to one or more managed object instances. This task can be performed using the menu option or a toolbar icon.
set_administrative_state[a]	Enables the user to place a managed object in the administrative state from the Map Presenter.
load_on_demand	This task must to assigned to all users being provided access to this application. It enables the application to pre-load appropriate information on the Map Presenter. Instructs the problem handler to pre-load the problems into the client controller when the application is started, based on the managed object selected by user from the Map Presenter.

a. The function related to this task is not available to the operator in the current release.

Table C-2 Tasks Associated with problem_application

Task Name	Description
get	This task must to assigned to all users being provided access to this application. 1. Enables the user to receive notifications from the FM Server. 2. Enables the user to query problem details from the FM Server.
set_trouble_ticket	Enables the user to submit one or more trouble -tickets to a problem. Enables the user to launch the trouble ticket application. The tt-id received through the above actions will be entered in the FM Server.
annotate	Enables the user to submit annotation notes to a problem.
own	Enables the user to own a problem.
disown	Enables the user to disown a problem that the user had earlier owned. To effectively use this task, the user must also be assigned the task `own`.
disown_any	Enables the user to disown problem conditions owned by other users.
discharge	Enables the user to discharge problem conditions that the user had earlier owned. To effectively use this task, the user must also be assigned the task `own`.
get_history	Enables the user to query all alarms associated with a selected problem. This is the Problem History function.
get_ne_history[a]	Enables the user to query both active and historical alarms associated with any selected network element.
get_related_alarms	Enables the user to query alarms related to a problem.

Table C-2 Tasks Associated with problem_application

Task Name	Description
load_on_demand	This task must to assigned to all users being provided access to this application. It enables the application to pre-load appropriate information on the Problem Presenter. Instructs the problem handler to pre-load the problems into the client controller when the application starts.

a. The function related to this task is not available to the operator in the current release.

Table C-3 Tasks Associated with outage_plan_application

Task Name	Description
get	This task must to assigned to all users being provided access to this application. 1. Enables the user to receive notifications from the FM Server. 2. Enables the user to query outage plan details on the FM Server.
manage	Enables the user to submit outage plans for one or more managed object instances.
restore	Enables the user to “manually” restore a managed object instance that is currently placed in outage. it is assumed that the same user had submitted the outage plan. Hence to use this task the user must also be assigned the task `manage`.
update	Enables the user to modify an existing outage plan that the user had created. Hence to effectively use this task the user must also be assigned the task `manage`.
restore_any	Enables the user to “manually” restore managed object that has been set on outage by another user.
update_any	Enables the user to modify outage plans submitted by other users.
load_on_demand	This task must to assigned to all users being provided access to this application. It enables the application to pre-load appropriate information on the Outage Plan Presenter. Instructs the application handler to pre-load the outage plans into the client controller when the application starts.

Table C-4 Tasks Associated with om_event_application

Task Name	Description
get	This task must to assigned to any user being provided access to this application. 1. Enables the user to receive notifications from the FM Server. 2. Enables the user to query OM events on the FM Server.
own	Enables the user to own an OM event.
disown	Enables the user to disown an OM event that the user had earlier owned. To effectively use this task, the user must also be assigned the task own.
disown_any	Enables the user to disown the OM event owned by any other user.
discharge	Enables the user to discharge an OM event that the user had owned earlier. To effectively use this task, the user must also be assigned the task own.
load_on_demand	This task must to assigned to all users being provided access to this application. It enables the application to pre-load appropriate information on the OM Event Presenter. Instructs the problem handler to pre-load the records into the client controller when the application starts.

Glossary

acknowledge Active messages in the iNOC console's Active Messages browser and Problem Presenter can be acknowledged by an administrator or operator or automatically after an action has been completed successfully.

action A response to a message triggered by an event.

admin group A system-defined user group. Users belonging to this group have supervisory rights over other users. Administrators can discharge and disown problem conditions owned by other users.

administrator A user who has privileges and responsibilities to configure and maintain a managed network.

agent A management component deployed on a system for the purpose of collecting events and injecting them as alarms into the management system. The agent performs basic event data collection and initial processing into a normalized alarm format.

alarm A message about an event that is collected from a network element or system and forwarded to the management system for processing.

annotation Text entered by operators, administrators, or automatically after actions that describe actions and tasks that have been applied to solve a given problem.

annotation server A user-supplied server that receives a request from an annotation node within a correlation circuit, performs some action, and returns a response to the annotate node. The action performed by the annotation server may involve information extracted from events in the circuit. The information returned is typically obtained external to the annotation server.

application handlers Applications that connect to the graphical user interface and are responsible for the accuracy and completeness of the data in the Client Controller.

application server Consists of a set of CORBA client applications, referred to as the application handlers, which interface between the topology server and the GUI Client. The application handlers supported are problem handler, map handler, outage plan handler, and OM event handler.

ARS Remedy Corporation's Action Request System, a network-based trouble ticketing and tracking system.

attributes Properties associated with a managed object class (See managed object class) and are registered under a registration identifier (See Registration ID).

attribute value assertion (AVA) The association of an attribute with a value, written in the form `attribute registration id = value`.

button panel A row of buttons either at the bottom or the right side of a window. Each button has a specific function. The function of a button is activated only when the button is highlighted. Menu greying indicates that the function is either not available or not applicable.

circuit See Correlation circuit.

CMIP (Common Management Information Protocol) A connection-oriented protocol that allows network elements, such as hosts, terminal servers, gateways, and management agents, to be manipulated via sophisticated messages.

CMISE The services defined for CMIP protocol are known as CMISE.

Client Controller The GUI Server process containing the object model and data and the views that manage the GUI Client presenters.

column based parser The parser type used for message classes that follow a single fixed column format.

component Physical objects contained within a network element. Components may or may not emit alarms.

component class A logical class type in an object model that can be contained under a network element class, termination point class, and other component classes. Objects in this class may or may not emit alarms.

composite event In ECS, a composite event consists of a structured aggregation of addressed component events, each of which may be a primitive event, a temporary event, or a composite event. A composite event may only exist within a correlation circuit.

connection See Link.

connection class A logical class type of an object model that can be contained under a network class. Objects in this class are used to link two managed objects via their termination point objects.

connection symbol A symbol on a network map that connects two map symbols.

containment hierarchy The rooted tree that is constructed by applying the relationship "is contained within" to the actual object instances (See Managed object instance). Lower level object instances are contained within object instances one level higher in the containment tree to which they are attached. The containment hierarchy follows the containment rules specified by the containment tree.

containment tree The rooted tree that is constructed by applying the relationship “can be contained into” object classes (See Managed object class). Lower level object classes can be contained within an object class one level higher in the containment tree to which they are attached. The containment tree specifies the containment rules by listing the classes of object instances that a particular object instance (of a particular object class) may contain.

CORBA (Common Object Request Broker Architecture)

A specification for objects to locate and activate one another in a distributed computing infrastructure.

correlation circuit In ECS, a collection of interconnected primitive and compound nodes configured to perform a filtering or correlation activity. Each correlation node is configured appropriately to the correlation requirement. The configuration includes the specification of the event types and the allowed transit delays for those events. A correlation circuit can be loaded into the ECS correlation engine.

correlation engine The ECS component that reads an input event stream, decodes the input events, performs the event correlation, encodes the output events, and returns the output events to the event stream. The

event correlation is as specified by one or more correlation circuits loaded into the correlation engine.

correlation node A set of customizable processing elements that facilitates the correlation of event storms in real time.

data collector A data collector receives messages emitted from network elements and forwards them to the agent.

data store In ECS, a component of the ECS engine that holds user-specified values for named data items. A correlation circuit may be associated with one of many data stores loaded into the correlation engine.

details An operation on the Problem Presenter that displays additional information about a network element emitting an alarm. The Problem Details panel displays the problem condition, owner, and FDN for a managed object.

device A piece of equipment that generates alarms when any of its components fail.

discharge An operation that removes a problem or event from the table area of table presenters. When you discharge a problem or event, it is removed for all users. Only operators who own a problem or event can discharge the problem or event.

disown An operation that releases a problem or event from its owner. When an event or problem is disowned, it appears in the table area of a Problem or OM Event Presenter for all users as unacknowledged. It is recommended that operators disown problems and events when they are not monitoring the network.

ECS See Event Correlation Services (ECS).

ECS circuit See Correlation circuit.

ECS Designer An ECS component that is used to create and test correlation circuits. It works in two modes: build and simulate. Must be purchased separately.

ECS Engine See Correlation engine.

Element Management System (EMS) Vendor- or device-specific components that provide device-specific interfaces for receiving events or monitoring the end network elements.

event correlation A process of filtering superfluous messages based on user-configured criteria.

Event Correlation Services (ECS) The HP Open View Event Correlation Services product, which uses correlation circuits and ECS engine to filter events.

explodable Map symbols that result in a submap upon double-clicking.

fact store A component of the ECS Engine that stores relationships among objects. Any two objects may be related using any user-defined relationship. The facts may be accessed at runtime by the ECDL expressions configured into the correlation node parameters.

Fault Management Server (FM Server) A topology server component that contains a representation of the underlying network. The FM Servers receive, store, and manage messages from the agents to which they are connected.

FDN (Fully distinguished name) The FDN uniquely identifies an object instance. The FDN is formed by concatenating all of the relative distinguished names (RDNs) for each object instance in the containment path from the root of the containment hierarchy to the base of the object instance that is being identified. The FDN is written as /RDN/RDN/.../RDN where each RDN is the RDN of the object instances along the containment path.

filter A condition that changes, suppresses, or redirects information to the topology GUI Clients.

FM Server See Fault Management Server.

GUI Client A set of GUIs that enable users to view topology-specific information. It connects to the topology server.

GUIDB See GUI database.

GUI database Used to store persistent graphical information, including user preferences and graphic layouts. Also known as the presentation database.

GUI Server A topology server component that is responsible for managing the display processes. It runs the application handlers that form the bridge between the FM Server process and the GUI Client.

high availability An optional package that enables the topology server to continue operations in spite of a single point of hardware or software failure.

history An operation in a Problem Presenter that displays all alarms associated with an active problem. When an operator owns or discharges a problem, all alarms associated with that problem become historical data and are not available with this operation.

host A server or workstation.

hostname The name of the server in the network.

information icon An icon that replaces secondary state icons when more than two secondary state icons are present for a single node.

installation A term used to describe an entire managed network, installed and configured with the topology server. Also referred to as a site.

interceptor An agent process dedicated to collecting alarms from a particular source. The logfile encapsulator collects alarms from log files. The opcmsg interceptor collects alarms injected using the opcmsg(3) API.

IP Stands for Internet Protocol. This is a datagram-oriented network layer protocol used by TCP and UDP protocols. Its main function is to route datagrams among nodes in different networks.

IPC Stands for InterProcess Communication. In this document, IPC specifically refers to the HP-UX IPC facility.

link Refers to the object that is represented by the connection symbol.

locate An operation in the table presenters that locates a problem, outage, or event in a managed network.

location A cluster of one of more machines offering telecom topology services. Each location contains only one topology server.

log file Files that store received messages emitted from network elements in a managed network, including a raw log, a report log, an unknown log, and a message class log.

managed object A logical or physical resource that can be managed. Every managed object is a member of a specific object class, whose members share the same set of attributes, operations that can be performed on them, notifications they can emit, and behaviors. With the topology server, managed objects are represented graphically by map symbols on Map Presenters.

managed object class (MOC) A group of managed object instances with the same or similar properties. An object class is registered under a unique registration ID, and is defined by a list of attributes, a list of naming attributes, a set of operations that it supports, and notifications or events that it can emit.

managed object instance (MOI) A representation of a specific occurrence of an object class to be managed. The MOI is addressed by a FDN.

management domain A logical grouping of network elements that is not dependent upon the physical boundaries of the network.

management server The central system from which all messages emitted from managed nodes are forwarded. The OV Operations software and relational database reside on the management server.

map presenter A type of GUI Client presenter that displays network topology information and status. It displays details of elements being monitored in a network. It also receives updates regarding the state and status of managed objects and network topology, and displays this information on a map.

map symbol A symbol on a network map that represents a managed object instance. An icon is assigned to each managed object

class, so managed object instances can be displayed in a Map Presenter. This assignment is performed by an administrator using the GUI Server configurator, the Admin Panel.

MC/ServiceGuard Allows the creation of a high availability cluster of HP 9000 Series 800 computers. A high availability computer system allows applications to continue in spite of a hardware or software failure.

message A structured, readable piece of information about a status, event, or problem related to a managed node.

message classes A network element can have multiple message classes. Basic details, such as message logging, must be specified for each message class. Configuration information includes message headers and trailers, message format, and message mapping to CMISE format.

naming attribute The attribute of an object class which distinguishes the object instances belonging to that class.

network class A logical class type of an object model that is the highest containment class of managed objects. Network classes, network element classes, and connection classes can be contained under a network class.

network element A piece of manageable telecommunications equipment that generates alarms when any of its components fail. For example, a network element can be a digital cross connect, an add-drop multiplexer, or a digital loop carrier.

network element class A logical class type of an object model that consists of network elements. This is the highest class of objects that can emit alarms.

NNM (Network Node Manager) An OpenView software product that discovers and manages a given IP and IPX network.

NOC (Network Operations Center) A place from which a network is supervised, monitored, and maintained.

node A connection point for data transmissions.

object A representation of a logical or physical entity or resource, or a group of such physical entities that exist in the network. Examples of objects are a network, a computer, an interface, and a process. Objects are represented graphically by the symbols that appear on submaps.

object instance See Managed object instance.

object model Enables the classification of devices in a monitored network into object classes, including network class, network element class, connection class, termination point class, and component class.

OM Event Presenter A type of GUI Client presenter that manages and displays non-alarm events generated by management applications. It contains the problem management functions: own, disown, discharge, and locate.

operation profile See profile.

Operation Profile Configurator A topology server GUI with which administrators configure operator IDs, roles, operation profiles, filters, and application domains.

Open Systems Interconnection (OSI) A systems management model that defines the rules for processing and transferring data over networks.

OSF/Motif GUI A graphical user interface standard that conforms to Open Software Foundation's recommendations.

outage plan Outage schedules to track when network elements are to be taken out of service from a monitored network.

outage plan presenter A type of GUI Client presenter that displays outage plans and schedules in tabular form. It contains the problem management functions: submit, modify, locate, and restore.

outstanding alarms Alarms, both new and acknowledged that have not yet been discharged.

OV Operations An OpenView software product that provides a generic framework for system, applications, and network management. Also known as OVO, VantagePoint Operations (VPO), and ITO.

OV DM TMN HP OpenView TMN Distributed Management Platform.

ovstart The program that starts up the OV DM TMN processes. This program is (normally) run automatically on system startup and can only be run by the superuser (i.e. UNIX System Administrator).

ovstop The program that stops OV DM TMN processes. This program is (normally) run automatically on system shutdown and can only be run by the superuser.

OVw HP OpenView Windows, an advanced graphical user interface designed to integrate network management and system management applications.

own An operation that assigns a problem or event to an operator. Owned problems and events appear in the table area of Problem and OM Event Presenters to all users as owned. After a problem or event is owned, a trouble ticket can be created to track the problem, or an operator can discharge it.

parser type The parser that is used for identifying the messages.

partition A set of managed objects grouped together based on physical characteristics or network technology. Each partition is associated with one location.

PDU (Protocol Data Unit) Used for the specification of association requests and responses.

presentation database See GUI database.

primary FM Server The FM Server from which system distribution rules are set. It is the server used to configure details for an installation or site.

problem The result of correlating multiple alarms with the same target object, probable cause, and specific problem and presenting them as a single instance to the user. Problems are higher level abstractions of groups of underlying alarms.

problem presenter A type of GUI Client presenter that displays problems in a tabular form. It contains the problem management functions: own, disown, discharge, locate, history, and details.

profile Collection of tasks, applications, capabilities, and responsibilities that can be assigned to a user.

radio buttons Radio buttons are typically used for setting states or modes. Depressed button state indicates that the parameter is selected.

raw alarms Alarm messages that are emitted from network elements in a managed network, and are not formatted or correlated.

RAV (Raw alarm viewer) A type of alarm viewer that displays real-time or query-based raw alarm messages for any network element in a managed network.

RDN (Relative Distinguished Name) List of Attribute Value Assertions (AVAs) of the naming attributes of the object class to which the object belongs. The RDN is written AVA, AVA, . . . , AVA.

registration ID A sequence of numbers used to uniquely identify such things as attributes and object classes.

regular expression based parser A parser type for message classes that can transmit messages in more than one format. These messages may or may not contain all the fields defined for the network element object class.

restore An operation in an Outage Plan Presenter that renews the status of network elements after an outage expires.

server A computer or network that provides service to other computers on the network (clients).

SNMP (Simple Network Management Protocol) The ARPA network management protocol used primarily for managing TCP/IP networks.

status propagation rules A list of statements that indicate to the system the parameters by which the status of the child class object in a network map hierarchy changes that of the parent class object. These rules are defined in the `sprules.conf` file in the `$FMSETC/share/newconf` directory.

submap A term used in the Map Presenter to refer to a view of a network map. For example, one submap may show all the nodes on a particular network, while another submap may show all the software subsystems of a particular node. The application or user that creates a submap determines the content of the submap.

submit An operation in an Outage Plan Presenter that enables users to create or modify an outage plan for a network element.

symbol A graphical representation of an object in a Map Presenter. An object can be represented by multiple symbols. A symbol has the characteristics symbol type, status, and label.

table area The rows and columns of table presenters where problems, outages, and events are displayed. It can be customized by users.

table presenters A classification for a type of GUI Client presenters that display problems, outages, and events in tabular form. Displayed attributes and operations differ for the different presenters. Table presenters include the Problem Presenter, Outage Plan Presenter, and the OM Event Presenter.

TCP (Transmission control protocol) A method or protocol used along with the internet protocol to send data in the form of message units between computers over the Internet.

termination point An object that connects to a link object.

termination point class A logical class type of an object model that can be contained under network element classes and other termination point classes. Objects in this class are used to form connections between managed objects via termination point objects.

topology server An OpenView software product that enables customers to manage telecom-specific networks. It integrates with OV Operations to provide a complete solution network management system.

TTS Trouble ticketing system.

UDP (User datagram protocol) A communications method or portal that offers a limited amount of service when messages are exchanged between computers in a network that uses the internet protocol. An alternative to TCP.

user handler A process responsible for handling topology server GUI logon requests and starting appropriate GUI and server processes and clients.

WAV (Web-based alarm viewer) A type of alarm viewer that displays problem information about a managed network in read-only format in a web browser. The WAV is useful for operators at remote sites.

X.733 A standard alarm format for OpenView Service Assurance for Convergent Services. X.733 specifies a well-defined set of alarm fields and values.

Index

A

adding
  CORBA logins, 65
Admin Panel
  bitmapmappings, 208
  color mappings, 208
  Data Type
    Action, 233
    Map, 193
    Mapping, 206
    Menubars, 222
    Presenter Types, 188
    Resources, 216, 217
    Roles, 188
    Site, 190
    Toolbars, 229
    Users, 186
  invoking, 184
  menu position, 226, 232
  navigation buttons, 185
  parameters, 224, 231
  system mappings, 207
admin profile, 166
alarm bell
  setting, 221
AMS
  archiving, 62
  logical backup/recovery, 62
  restoring, 62
application
  tasks, 169
application domain
  viewing tasks associated with an application, 169
archiving
  AMS, 62
authentication
  add, 178

B

backup
  CORBA, 57
blinking objects
  configuring, 215
  expiry time, 216

C

checking
  IORs, 251
checking error log
  oemflinkmap
    ovdumplog, 128
  oemflinkmap.errout, 128
checking logs
  for oemflinkmap, 128
Collection managers
  setting trace levels, 261
color mappings, 208
colors
  Color Chooser, 219
  defined, 219
  Font Chooser, 220
CORBA
  adding logins, 65
  backup/recovery, 57
  data files, 54
  database administration, 55
  database concepts, 55
  database time synchronization, 63
  directory backups, 57
  errors, 253, 255
  listing hosts, 34
  listing logins, 65
  managing databases, 54
  managing message logs, 255
  managing notifications, 50

managing unified services, 51
managing users, 64
removing logins, 65
searching log messages, 253
starting, 41
stopping, 41
troubleshooting, 251

D

databases
CORBA, 54
CORBA backup/recovery, 57
domain
managing domains using the Map Presenter, 125

E

environment variable
ILT_MAXOBJECTS_PER_MAP, 38
ILTLOGLEVEL, 262, 263
MIS_TRACE_LEVEL, 260, 261
OVTRACE_LEVELS, 261
TOP_TRACE_LEVEL, 261
error logs, 255
errors
CORBA, 253
ORBPlus, 255
event type, 157

F

file format
import file, 121
filter, 147
add, 158
copy, 163
define, 157
delete, 163
modify, 162
fixing database inconsistencies
problem database, 35
FM Server
activating trace logs, 260
deactivating trace logs, 261
fmsalmbackup, 74, 76
fmsalmrecover, 35
fmsalmrestore, 76
fmseccfgupd, 239, 240
fmslogconadmin, 239
fmsomeventbackup, 75, 77
fmsomeventrestore, 77
fmsomgen, 123
fmsopcfg, 138, 143
fmsoutagebackup, 74
fmsoutagerestore, 77
fmsstart, 32
fmsstatus, 243
fmsstop, 37
fmssysconfig
utility, 85
fmstopofix, 265
usage, 265
fonts
defined, 220
setting, 221–??
FORCE_TELCO_GUI_LOGIN_AUTHENTICATION, 178
format
date for utilities, 71
function
Add to Domain, 125
oemflinkmap, 127
remove from domain, 125
remove fromdomain, 126
FunctionToIconMapping, 201

G

GUI Client
  activating trace logs, 262, 263
  deactivating trace logs, 262, 263
GUI Server
  activating trace logs, 262
  deactivating trace logs, 262
guidbmocsym, 127, 205
guidbrestore, 80
guisadmin, 184
guisstart, 38
guisstatus, 39, 243
guisstop, 39

H

hosts
  listing, 34

I

ILT_MAXOBJECTS_PER_MAP, 38
IltCoMo2MapObjClassMapping, 204
ILTLOGLEVEL, 262, 263
import file
  field description, 121
iNOC Console, 133
  configure access, 137
IORs
  checking, 251

L

link object submap
  creating, 126
listing
  CORBA logins, 65
  hosts, 34
Listing FM hosts, 34
Local Form Probable Cause, 212
locked sessions
  accessing, 248
log messages
  for CORBA, 253
login_telco_gui.bat, 179
login_telco_gui.ksh, 178
logins
  adding, 65
  listing, 65
  removing, 65
logs
  managing, 255

M

managed object domain, 146, 148
  add, 148
  delete, 150
management domain group, 146, 151
  add, 152
  delete, 156
  modify, 155
management domains
  managed object, 146, 148
    adding, 150
    deleting, 150
    function, 148
  management domain group, 146
    adding, 152
    copying, 155
    deleting, 156
    modifying, 155
managing
  CORBA databases, 54

CORBA users, 64
message logs, 255
notifications, 50
unified services, 51
map
copying for another user, 192
symbols, 196
map object classes
defined, 196
Map Presenter
adding objects to domains, 125
managing domains, 125
removing objects from domain, 125, 126
MapOC
PictorialNEClass, 197
ShapeNEClass, 197
SymbolicNEClass, 196
MapOCs
predefined, 198
mapping
acknowledged icons, 211
maps
icons, 201
MOC to MapOC, 204
MOC to MapOC mapping, 205
menu
adding items, 227
Message Logging Service, 253
message logs
managing, 255
MIS_TRACE_LEVEL, 261

N

Notification Service, 50
NSM processes
setting trace levels, 260, 261

O

Object Locator
backup/recovery, 57
ODBC, 56
oemfadm user, 137
oemferr, 244
oemflinkmap, 127
oemflinkmapimport, 127
oemfpasswd, 85
oemfstart, 43
oemfstatus, 243
oemfstop, 43
opc_adm user, 135
operation profile, 134, 146, 147, 165
add, 166
add applications, 168
filter, 172
mamagement domain group, 170
managed object domain, 171
time restriction, 176
user, 174, 177
work schedule, 175
Operation Profile Configurator, 137
operation profiles
associating applications, 168
associating MDG managment domains, 170
associating MO management domains, 171
associating problem filters, 172, 173
associating users, 174, 177
Operation Profiles Configurator
about, 138
Oracle

use in CORBA, 55
ORB Plus
error messages, 255
OS user
add, 143
ov_amsuser_db, 54, 62
ovcorba
database file locations, 58
database tablespaces, 59
ovcorba_admin, 41
ovcorba_db_config, 54
ovcorba_orb, 41
ovdumplog, 253, 254, 255
fmsomgen, 124
oemflinkmapimport, 127
ovlerch, 251
ovlogalarm, 255
ovlogin, 64, 65
ovlognotifs, 253, 254, 255
ovlogrollover, 255
ovlogsev, 256
ovlogsize, 256
ovlslogs, 256
ovlstypes, 256
ovnsls
database file locations, 58
database tablespaces, 59
ovnsls_admin, 41
OVO operator GUI, 134
OVO user
add, 142
delete, 144
ovsiteinfo, 34
ovsync_amsuser, 54
ovtdbadmin, 54
ovuser, 64, 65

P

Parameters dialog, 224, 231
physical backup
GUIDB, 80
PictorialNEClass, 197
Position dialog, 226, 232
probable cause, 157
local form mapping, 212
M3100, 161
X.721, 161
problem filter
adding, 159
copying, 163
deleting, 163
modifying, 162
profile, 146

R

RDBMS, 55
recovery
CORBA, 57
Relational Database
Management Server, 55
removing
CORBA logins, 65
restoring
AMS, 62

S

sdbmcp, 57, 58
Session
Set home session, 227
Set Alarm Alert, 221
setting fonts, 221–??
severity, 157
ShapeNEClass, 197
shutting down
all servers together, 43
FM Server, 37
specific problem, 157

starting
  CORBA, 41
starting up
  all servers together, 43
  FM Server, 32
stopping
  CORBA, 41
SymbolicNEClass, 196
syslog, 255
System Configuration Utilities (*sysconfig), 24

T

Telco iNOC application, 140
Telco_Op profile, 135, 139
telco_op user, 135, 139
time synchronization, 63
time zone, 176
Topodb APIs
  setting trace levels, 261
topologin.user, 140, 178
topology GUI, 134
  login, 135
topology GUI login
  change profile, 179
topology GUI user
  add, 143
  delete, 145
topoprofile.user, 141, 178
Trace Log
  activate FM trace logs, 260
  activate GUIC trace logs, 262, 263
  activate GUIS trace logs, 262
  deactivate FM trace logs, 261
  deactivate GUIC trace logs, 262, 263
  deactivate GUIS trace logs, 262
Trader
  backup/recovery, 57
troubleshooting
  CORBA, 251

U

Unified Services, 51
user, 134
User Bank, 142
User Profile Bank, 142
User Service, 64
  logical backup/recovery, 62
Utilities
  date format, 71
Utility
  fmsalmbackup, 74, 76
  fmsalmrecover, 35
  fmsalmrestore, 76
  fmseccfgupd, 239, 240
  fmslogconadmin, 239
  fmsomeventbackup, 75, 77
  fmsomeventrestore, 77
  fmsomgen, 123
  fmsopcfg, 138
  fmsoutagebackup, 74
  fmsoutagerestore, 77
  fmsstart, 32
  fmsstatus, 243
  fmsstop, 37
  fmssysconfig, 85
  fmstopofix, 265
  guidbmocsym, 127
  guidbrestore, 80
    guidbrestore, 79
  guisadmin, 184
  guisstart, 38
  guisstatus, 39, 243
  guisstop, 39
  oemferr, 244

oemflinkmap, 126, 127
oemflinkmapimport, 127
oemfpasswd, 85
oemfstart, 43
oemfstatus, 243
oemfstop, 43
ovdumplog, 127, 128
  fmsomgen, 124
System Configuration
    (*sysconfig), 24

W

work schedule, 175

Index