

Open System Management Framework (OSMF)

for HP-UX

Managed Object Agent
(MOA) Option

Administrator's Guide

This guide provides information necessary to install, register, and run the OSMF Managed Object Agent option (referred to also as the "OSMF MOA" throughout this manual) on your OSMF workstation. The information in this manual applies to the MOA software having the software version number listed below, unless accompanied by a "Manual Updating Changes" package indicating otherwise.

SOFTWARE VERSION NUMBER: 1.3

This document explains attaching the mounting brackets to the 58502A.

For assistance, contact:

Symmetricom, Inc.
2300 Orchard Parkway
San Jose, CA 95131-1017

U.S.A. Call Center:
888-367-7966 (from inside U.S.A. only – toll free)
408-428-7907

U.K. Call Center:
+44.7000.111666 (Technical Assistance)
+44.7000.111888 (Sales)
+44.1604.586740

Fax: 408-428-7998

E-mail: ctac@symmetricom.com

Internet: <http://www.symmetricom.com>

Warning Symbols That May Be Used In This Book



Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.



Indicates hazardous voltages.



Indicates earth (ground) terminal.



or



Indicates terminal is connected to chassis when such connection is not apparent.



Indicates Alternating current.



Indicates Direct current.

Contents

1	Introduction and General Information	
1.1	In This Guide.....	1-2
	Topics Included in This Guide	1-2
	Naming Conventions Used in this Guide	1-2
1.2	Description of the MOA.....	1-2
	Event Forwarding	1-4
	OSMF MOA Processes	1-5
1.3	Materials Provided	1-6
1.4	Responsibilities of the System Administrator	1-6
1.5	System Requirements.....	1-6
	Software Requirements	1-6
	Hardware Requirements	1-6
	Oracle Database Requirements	1-6
1.6	Prerequisite Knowledge	1-6
1.7	Related Documentation.....	1-7
2	Installation	
2.1	In this Chapter	2-2
2.2	Overview	2-2
2.3	Installing the HP OSMF MOA.....	2-3
2.4	Starting the MOA	2-4
2.5	Stopping the MOA	2-5
3	Accessing the MOA from a Manager	
	In this Chapter.....	3-2
	Overview.....	3-2
	Configuring the ORS Nodes	3-2
A	Messages Supported and Implemented	
A.1	Messages implemented between REGB of OSMF-MOA‡ GNCE of INM.....	A-2
	A.1.1 xdr_regbgnce_accept_gnce_session_t	A-2
	A.1.2 xdr_regbgnce_keep_alive_t	A-2
	A.1.3 xdr_regbgnce_send_mopc_info_t	A-2
	A.1.4 xdr_regbgnce_send_ne_type_info_t	A-4
A.2	Messages implemented between REGD of OFMF-MOA‡ COC of INM.....	A-5
	A.2.1 xdr_regdcoc_accept_coc_session_t	A-5
	A.2.2 xdr_regdcoc_accept_copc_registration_t	A-5
	A.2.3 xdr_regdcoc_deny_copc_registration_t	A-6
	A.2.4 xdr_regdcoc_keep_alive_t	A-7

A.3 Messages implemented between mainProcess(UOC) of OSMF-MOA‡ COC of INMA-8

- A.3.1 xdr_uoccoc_clear_alarm_t A-8
- A.3.2 xdr_uoccoc_close_moc_session_t A-8
- A.3.3 xdr_uoccoc_deenroll_ne_t A-8
- A.3.4 xdr_uoccoc_enroll_ne_t A-9
- A.3.5 xdr_uoccoc_keep_alive_t A-10
- A.3.6 xdr_uoccoc_ne_counts_inaccurate_t A-10
- A.3.7 xdr_uoccoc_open_moc_session_t A-11
- A.3.8 xdr_uoccoc_raise_alarm_t A-11
- A.3.9 xdr_uoccoc_send_ne_list_t A-14
- A.3.10 xdr_uoccoc_synchronize_t A-14
- A.3.11 xdr_uoccoc_update_copc_info_t A-14
- A.3.12 xdr_uoccoc_update_mopc_activity_t A-15
- A.3.13 xdr_uoccoc_update_ne_counts_t A-16
- A.3.14 xdr_uoccoc_update_mopc_support_t A-16

A.4 Messages supported between GNCE of INM‡ REGB of OSMF-MOA A-17

A.5 Messages supported between COC of INM‡ REGD of OSMF-MOA..... A-18

A.6 Messages supported between COC of INM‡ mainProcess(UOC) of OSMF-MOA A-19

1.1 In This Guide

This *55452A OSMF Software OSMF MOA System Administrator's Guide* provides information necessary to install, manage, and maintain the OSMF MOA software (referred to also as the MOA software or MOA). It makes the assumption you have HP-UX, HP OpenView, and OSMF software experience.

1.1.1 Topics Included in This Guide

This guide includes the following topics:

- A description of the MOA software.
- Minimum system resources required to run the MOA.
- Responsibilities of the system administrator.
- Basic system administration tasks, such as installing the MOA, and starting and stopping it.

1.1.2 Naming Conventions Used in this Guide

This guide uses several names to refer to the same objects. The usage of multiple terms for the same object helps define the relationship between objects in different situations. To establish naming conventions, the objects and their corresponding names are listed in the following table:

Object	Names Used for the Object
55452A open synchronization management framework software	OSMF software for HP-UX, OSMF software, OSMF
HP-UX Workstation	OSMF workstation, OSMF system, host system, element manager
TMN Upper-Layer Network Management System	upper-layer management system, upper-layer network manager, upper-layer manager, network manager
OSMF Managed Object Agent	OSMF MOA, MOA
Nortel's Integrated Network Manager	INM

1.2 Description of the MOA

The 55452A OSMF software, integrates HP OpenView Windows with an Oracle database and many other modules to facilitate management of SSUs through a user-friendly interface. A higher-layer network management system which might need to communicate with SSUs must do so over an XDR interface using a managed object agent (MOA) running on the OSMF platform. The MOA thus used is the OSMF MOA.

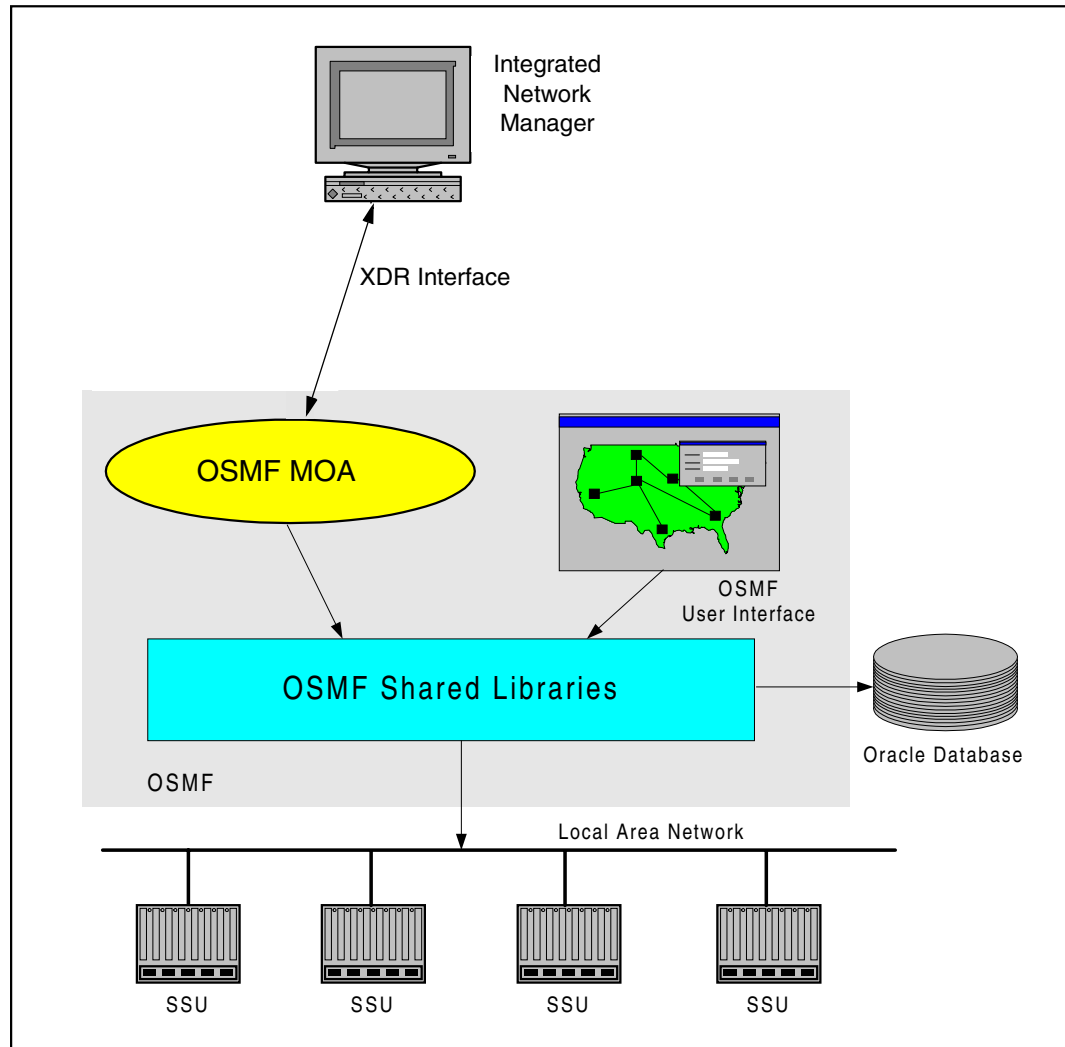
The OSMF MOA translates (as required) the information (commands, data, status information, events, etc.) that passes between the OSMF software and an upper-layer integrated network manager (INM). The MOA provides the following functions:

Description of the MOA

Issue 1

- Allows the INM to access up to 1000 network elements (SSUs).
- Enables the INM to perform NE and alarms surveillance.
- Translates SSU-generated TL/1 events to XDR format so that the INM can process them.

The following figure shows that within the OSMF application, the MOA interacts with other modules by using the OSMF shared libraries.

**The Information Model and OSMF**

The OSMF software provides a communication interface library which is used for communication between the OSMF MOA and Nortel's INM.

The main function of the MOA involves forwarding of events (event forwarding). Event forwarding involves the processing and forwarding of events from network elements and the system itself to the upper-layer INM.

1.2.1 Event Forwarding

During Event Forwarding, the OSMF database does the following in the order shown:

- Receives TL1 events from the Event Receiver (sveht) or the user interface.
- Processes them.
- Notifies the MOA of the events via triggers to an Event Dispatcher.

The MOA then retrieves the events, processes them, and sends acknowledgments back to the database.

There are a variety of events that can be forwarded to the MOA from the database. Events can be SSU generated, OSMF generated, or system related. All these events are converted to XDR format by an event translator within the MOA and sent to Nortel's INM.

Events can be classified under the following three categories:

SSU Generated Events. These events are generated asynchronously by the SSU. They include both alarmed and non-alarmed events which originate from the SSU as TL1 messages. It is the responsibility of the MOA to convert these TL1 events to XDR format based upon the mapping specified in the compliance table of messages supported/implemented between the OSMF MOA and Nortel's INM, in Appendix A.

Operator Initiated Events. Whenever an operator uses the Configuration->Modify form to modify an attribute in the SSU, the MOA is notified of this change via an event. These operator initiated events include the following:

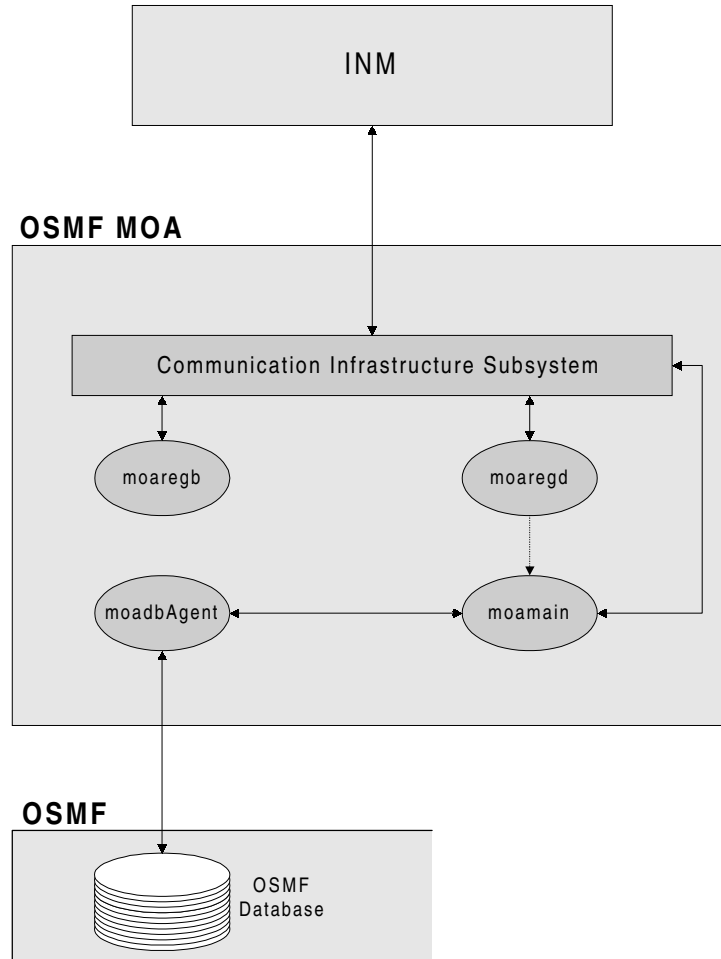
- Changes to an SSU's operational state (managed, unmanaged).
- Addition of a new SSU.
- Deletion of an SSU.

Administrative Events. These events could be either synchronous or asynchronous and are generated as a result of administrative changes within OSMF. The events that are currently supported in this category are as follows:

- Heartbeat failure for an SSU.
- Recovery from a heartbeat failure.
- Alignment failure for an SSU (configuration mismatch).
- Recovery from alignment failure.
- Inventory mismatch.

1.2.2 OSMF MOA Processes

The OSMF MOA consists of four processes. These processes use the OSMF communication interface library for communicating with the INM. The four OSMF MOA processes are shown in the following figure and explained in the text that follows:



MOA Registration Browser (moaregb). This process provides information about the OSMF MOA's span of control to the INM.

MOA Registration Daemon (moaregd). This process is responsible for registering the INM with the OSMF MOA for NE and Alarm details provisioning.

MOA Main Process (moamain). This process is responsible for sending the alarms and events and NE details of OSMF to the INM.

MOA Database Agent (moadbAgent). This process monitors the OSMF database and provides the relevant database information to the MOA main process. This process does not communicate with the INM.

1.3 Materials Provided

The OSMF MOA Option includes the materials listed in the following table:

Description	Qty
Software on DAT tape	1
55452A OSMF Software OSMF MOA System Administrator's Guide (Documentation)	1

1.4 Responsibilities of the System Administrator

The system administrator is responsible for maintenance of the OSMF system and the MOA. This requires the system administrator to understand the workstation hardware and all of the software installed.

The main responsibilities of the system administrator are as follows:

- Installing the MOA software.
- Starting and stopping the MOA.

1.5 System Requirements

The hardware and software described in the following sections is required to run the MOA.

1.5.1 Software Requirements

In addition to the software requirements detailed in the section entitled "System Requirements" in the *55452A OSMF Software System Administrator's Guide*, the MOA requires that the OSI Transport Services product, HP OTS/9000, be installed on the workstation running the MOA.

1.5.2 Hardware Requirements

The MOA runs on the same workstation on which the OSMF product runs. Refer to the *55452A OSMF Software System Administrator's Guide* for information on the hardware required to run the OSMF software.

1.5.3 Oracle Database Requirements

Version 1.3 of the MOA is designed to work with release 7.3.3.XX of the Oracle database.

User Class Capabilities (General)

User	Users in this user class can view the network map as well as view the synchronization management information in the Alarm Viewer, Event History Log, Performance Monitor, Inventory, and configuration dialog boxes. Users in this class can also create and print event and performance reports.
Administrator	Users in this user class have all of the capabilities of the “User” class, and can additionally modify the inventory database and configuration of NSUs.
Security	Users in this user class have all of the capabilities of the “User” and “Administrator” classes, and can modify the network topology, manage or unmanage NSUs, and configure management domains for users.

1.6 Prerequisite Knowledge

The procedures in this manual assume that the system administrator has HP-UX, HP OpenView, and OSMF software experience. This knowledge is useful, and in some cases necessary when performing the tasks in this manual.

1.7 Related Documentation

Some procedures in this guide may refer you to procedures or other information that is contained in the *55452A OSMF Software System Administrator's Guide*.

This page intentionally left blank.

2.1 In this Chapter

The procedures in this chapter explain how to do the following:

- Install the OSMF MOA Software (referred to as the MOA).
- Start the MOA.
- Stop the MOA.

The procedures in this chapter assume that the following is true:

- You have at least the minimum standard hardware required for an OSMF system (see the section titled “System Requirements” in the *55452A OSMF Software System Administrator’s Guide*).
- You have HP OpenView DM 4.21, HP OTS/9000, and the Oracle software installed on your workstation.
- You have sufficient knowledge or experience with HP-UX, the OSMF software, and the system hardware to be able to follow these installation instructions.

NOTE

OSMF software, version 3.0 with patch 002 is a prerequisite for version 1.3 of the MOA. Before attempting to install the MOA, ensure that the OSMF software, version 3.0 with patch 002 is installed on the workstation.

2.2 Overview

The installation program, *swinstall*, does the following:

- Copies the MOA’s files to their respective directories.
- Creates a special user account called ‘q3agent.’
- Modifies the database to flag the availability of the MOA option for OSMF.
- Registers the MOA with HP OpenView.

NOTE

Registration is the way the MOA tells HP OpenView that it exists:

- Registration provides information that enables HP OpenView to start or stop the MOA.
- Registration updates the object database with information pertaining to the objects that the MOA will manage. This information is made publicly available to other management stations by the Object Registration Service (ORS) daemon. For more information on object registration please refer to the *HP OpenView TMN Administrator’s Guide*.

OpenView’s *ovaddobj* command is used to add the following four LRF files to the ORS database.

```
/etc/opt/OV/share/lrf/moaregb.lrf  
/etc/opt/OV/share/lrf/moaregd.lrf  
/etc/opt/OV/share/lrf/moamain.lrf  
/etc/opt/OV/share/lrf/moadbAgent.lrf
```

2.3 Installing the OSMF MOA

To install the MOA, use the Software Distributor, as described in the following procedure.

- 1 Log in as **root**, and type the following command to source the OSMF environment (this command must be run from the Korn shell [ksh]):

```
. /opt/smartview/config/env/sv-ux.env
```

- 2 Type the following command to stop HP OpenView:

```
svstop
```

- 3 Insert the OSMF MOA installation tape into the local system tape drive.

NOTE

You can run “swinstall” directly as an alternative to running it from SAM. If you choose to do this, it is only necessary to perform steps 6 through 11.

- 4 Type the following command to start the System Administration Manager (SAM):

```
/usr/sbin/sam
```

- 5 Double click on **Software Management**.

- 6 Double click on **Install Software to Local Host**.

- 7 Click on **Source Depot Type**, then select **Local Tape**.

- 8 Click on **Source Depot Path**, select the path to the local tape drive, then click **OK**.

Typically, the path to the local tape drive will be /dev/rmt/0m.

- 9 In the Software Selection window, highlight **HP55452CF**, and click on **Actions>Install (analysis)**.

Install analysis begins automatically, and it normally completes with a status of “Ready.” If a status other than “Ready” is indicated, click the **Logfile...** button in the “Install Analysis” dialog box for information about the problem.

- 10 After install analysis completes, click **OK**, and then click **Yes** in the confirmation window that appears.

After “Yes” is clicked, the execution phase of the install process will start. This phase should complete with no errors.

- 11 When the status shows “Completed” and “Percent Complete=100%,” click **Done**.

- 12 Click **File>Exit** to exit Software Distributor.

- 13 Click **File>Exit SAM** to end the software installation process and exit SAM.

2.4 Starting the MOA

After you have installed the MOA software as described in the above procedure, it is automatically started along with other OSMF daemons when the `svstart` command is executed. Use the following procedure to start the MOA:

- 1 Log in as **root**, and type the following command to source the OSMF environment (this command must be run from the Korn shell [ksh]):

```
. /opt/smartview/config/env/sv-ux.env
```

- 2 Type the following command to start the HP OpenView processes:

```
svstart
```

This command starts all processes, including the MOA. After all processes are spawned, the HP-UX root prompt returns.

NOTE

If the HP OpenView and OSMF processes are already running, the command **moastart** can be used to start the MOA independently.

- 3 Type the following command to ensure that the OSMF software-related processes and the MOA-related processes are up:

```
svstatus
```


The “svstatus” command outputs a status report to the HP-UX window. The following shows what the report contains if all OSMF software processes and MOA-related processes are up:

```
OSMF, Release 3.0 (build build_no)

The OSMF environment is set
-----

Openview is up
-----

Status of OSMF daemons ...
-----

Alignment Daemon:           up
HeartBeat Daemon:          up
Event Receiver Daemon:     up
Event Alert Daemon:        up
OVW Communication Daemon:  up
Status Sync Daemon:        up
GPS Agent Daemon:          up

MOA moaregb Daemon:        up
MOA moaregd Daemon:        up
MOA moamain Daemon:        up
MOA moadbAgent Daemon:     up

Replication Option is not installed
-----
```

2.5 Stopping the MOA

The MOA is automatically stopped along with other OSMF daemons when the *svstop* command is executed.

NOTE

If the HP OpenView and OSMF processes have already been stopped, the command **moastop** can be used to stop the MOA processes independently.

- 1 Log in as **root**, and type the following command to source the OSMF environment (this command must be run from the Korn shell [ksh]):

```
. /opt/smartview/config/env/sv-ux.env
```

- 2 Type the following command to stop the MOA-related processes:

```
moastop
```

This command stops the four MOA-related processes. After all processes have terminated, the HP-UX root prompt returns.

This page intentionally left blank.

3.1 In this Chapter

In a typical situation, the upper-layer manager and MOA reside on separate machines, possibly across a distributed network. In these situations, the upper-layer manager must know the following:

- Where the MOA exists
- The managed objects and instances that are under the MOA's control
- How the MOA can be reached.

The HP OpenView Object Registration Service (ORS) can be configured to satisfy the above three requirements. This chapter provides information on configuring HP OpenView platforms at both ends of a distributed network connection so that the upper-layer manager and MOA can communicate.

NOTE

A detailed description of the ORS can be found in the *HP OpenView Telecom DM Administrator's Guide*.

3.2 Overview

Every machine that runs the HP OpenView DM platform has a daemon called the `orsd` daemon. The `orsd` daemon automatically starts whenever the DM platform-related processes start. When many machines participate in a distributed management environment, one of the machines (or, nodes) must be configured as a master ORS node. All other nodes participating in the distributed management environment must be configured as slave ORS nodes. The master ORS node maintains a global repository (database) of all the managed objects that are controlled by each slave node. Thus, when an upper-layer manager wants to manage a specific instance of a managed object, it first queries the master ORS node to determine where the corresponding MOA resides, and then sends a request it.

3.3 Configuring the ORS Nodes

The following procedure configures the upper-layer manager's machine as the master ORS node and the MOA's machine as a slave ORS node.

NOTE

The following procedure must be performed as superuser (root).

- 1 Type the following command on the upper-layer manager's machine to verify that it is configured as a master ORS node:

```
ovorsutil -h
```

By default, when an `orsd` daemon starts up for the first time, it configures the node it is running on as a master ORS node. This should be reflected in the output of the `ovorsutil -h` command, which should appear similar to the following:

ovorsutil: The database contains the following master/slave hierarchy:

```
Master ORS node is: OVDM,ses0,tp0,manager_address
```

```
The following nodes are registered as slave ORS entries:
```

```
The total number of slave entries found is 0.
```

The string `OVDM,ses0,tp0,manager_address` represents the RFC1006 presentation address of the master ORS node. This string must match the presentation address found in file `"/etc/opt/OV/share/conf/local_p_addr"` on the upper-layer manager's machine.

- 2 View the upper-layer manager's `"/etc/opt/OV/share/conf/local_p_addr"` file to verify that the presentation address in it matches the presentation address in the output of the **ovorsutil -h** command.
- 3 If there is a mismatch in the addresses, type the following command on the upper-layer manager's machine:

```
ovorsutil -m OVDM,ses0,tp0,manager_hostname
```

Where *manager_hostname* is the hostname of the upper-layer manager's machine. This command directly sets the upper-layer manager's machine to be the master ORS node. When the command executes successfully, the message "ovorsutil: the change of master node has been requested" will be displayed.

- 4 Type the following command on the MOA's machine to configure it as a slave ORS node:

```
ovorsutil -m master_hostname
```

Where *master_hostname* is the hostname of the master ORS node, which is the upper-layer manager's machine. After this command is executed, the two machines will attempt to synchronize their ORS databases, during which messages that are similar to the following will be displayed:

```
ovorsutil: trying to contact master node OVDM,ses0,tp0,manager_address.
ovorsutil: contact successful, proceeding to request the change.
ovorsutil: the change of master node has been requested.
```

- 5 Type the following command on the upper-layer manager's machine:

```
ovorsutil -h
```

The output of this command should now list the MOA's machine as one of the machines registered as a slave ORS entry.

- 6 Verify that the ORS databases have been synchronized by typing the following command on the upper-layer manager's machine:

```
ovorsutil -d q3hp
```

If the MOA had been installed successfully, the output should show a listing of all the managed objects pertaining to the OSMF MOA, along with the presentation address of the MOA's machine.

NOTE

The MOA can now be accessed from the upper-layer manager that resides on the master ORS node.

A

Messages Supported and Implemented

A.1 Messages implemented between REGB of OSMF-MOA → GNCE of INM

A.1.1 *xdr_regbgnce_accept_gnce_session_t*

A.1.1.1 Supported fields

- regb_major_sw_version
- regb_minor_sw_version

A.1.1.2 Unsupported fields

Nil.

A.1.1.3 Possible values for supported fields

regb_major_sw_version	6
regb_minor_sw_version	0

A.1.2 *xdr_regbgnce_keep_alive_t*

A.1.2.1 Supported fields

Nil.

A.1.2.2 Unsupported fields

Nil.

A.1.2.3 Possible values for supported fields

Nil.

A.1.3 *xdr_regbgnce_send_mopc_info_t*

A.1.3.1 Supported fields

- mopc_name
- mopc_function
- mopc_sw_load
- mopc_activity
- mopc_availability
- copc_name
- copc_ip_adress
- coc_tcp_port_number

**Messages implemented between REGB of OSMF-MOA and GNCB
of INM**

- mopc_support_bits
- controller_type_name
- primary_serial
- controller_type

A.1.3.2 Unsupported field

- backup_serial(NULL)
- mopc_release_info

A.1.3.3 Possible values for supported fields

mopc_name	Hostname of the OSMF-MOA platform.
mopc_function	xdr_primary
mopc_sw_load	“OSMF-UX 3.0”

mopc_activity	xdr_active
mopc_availability	xdr_available OR xdr_already_registered
copc_name	The name of an already connected INM.
copc_ip_address	The IP address of an already connected INM.
coc_tcp_port_number	The port number of an already connected INM.
mopc_support_bits	01
controller_type_name	“OSMF-MOA”
primary_serial	“Primary OSMF”
controller_type	xdr_controller_hp_smartview_moa(10)

A.1.4 xdr_regbgnce_send_ne_type_info_t

A.1.4.1 Supported fields

- ne_type_info
- ne_type
- ne_label

A.1.4.2 Unsupported fields

- ne_icon_bitmap

A.1.4.3 Possible values for supported fields

ne_type_info	“HP 55400 SSU”
ne_type	90
ne_label	“SSU”

Messages implemented between REGD of OFMF-MOA → COC of INM

A.2 Messages implemented between REGD of OFMF-MOA → COC of INM

A.2.1 *xdr_regdcoc_accept_coc_session_t*

A.2.1.1 Supported fields

- regd_major_sw_version.
- regd_minor_sw_version

A.2.1.2 Unsupported fields

Nil.

A.2.1.3 Possible values for supported fields

regd_major_sw_version	4
regd_minor_sw_version	0

A.2.2 *xdr_regdcoc_accept_copc_registration_t*

A.2.2.1 Supported fields

- mopc_name
- mopc_function
- mopc_sw_load
- copc_name
- copc_ip_address
- coc_tcp_port_number
- mopc_release_info
- accepted_major_sw_version
- accepted_minor_sw_version

A.2.2.2 Unsupported fields

- controller_type

A.2.2.3 Possible values for supported fields

mopc_name	Hostname of OSMF-MOA
mopc_function	xdr_primary

mopc_sw_load	“OSMF-UX 3.0”
copc_name	The name of an already connected INM to OSMF-MOA.
copc_ip_address	The IP address of an already connected INM to OSMF-MOA.
coc_tcp_port_number	The TCP port number of an already connected INM to OSMF-MOA.
mopc_release_info	“OSMF-MOA 1.3”
accepted_major_sw_version	7
accepted_minor_sw_version	0

A.2.3 xdr_regdcoc_deny_copc_registration_t

A.2.3.1 Supported fields

- error_code
- copc_name
- copc_ip_address
- coc_tcp_port_number

A.2.3.2 Unsupported fields

Nil.

A.2.3.3 Possible values for supported fields

error_code	xdr_invalid_application
copc_name	The name of an already connected INM to the OSMF-MOA.
copc_ip_address	The ipAddress of an already connected INM.
coc_tcp_port_number	The listening port number of already Connected.

A.2.4 xdr_regdcoc_keep_alive_t

A.2.4.1 Supported fields

Nil.

A.2.4.2 Unsupported fields

Nil.

A.2.4.3 possible values for supported fields

Nil.

COC of INM

A.3 Messages implemented between mainProcess(UOC) of OSMF-MOA → COC of INM

A.3.1 *xdr_uoccoc_clear_alarm_t***A.3.1.1 Supported fields**

- ne_id
- alarm_id

A.3.1.2 Unsupported fields

- provision_change(Is set to FALSE)

A.3.1.3 Possible values for supported fields

ne_id	This value will be mapped to nes.idnt_ov_objt field of OSMF database
alarm_id	This value is mapped to current_alarms.event# of OSMF database

A.3.2 *xdr_uoccoc_close_moc_session_t***A.3.2.1 Supported fields**

Nil.

A.3.2.2 Unsupported fields

Nil.

A.3.2.3 Possible values for supported fields

Nil.

A.3.3 *xdr_uoccoc_deenroll_ne_t***A.3.3.1 Supported fields**

- ne_id

A.3.3.2 Unsupported fields

**Messages implemented between mainProcess(UOC) of OSMF-
MOA‡ COC of INM**

Nil

A.3.3.3 Possible values for supported fields

ne_id	This value is mapped to nes.idnt_ov_objt of OSMF database
-------	---

A.3.4 xdr_uoccoc_enroll_ne_t

A.3.4.1 Supported fields

- ne_id
- ne_type
- ne_name
- counts_accurate
- ne_tx_rate
- ne_shelf_type
- ne_support_bits
- ne_counts
- protection_type
- protection_summary
- valid_pm_stats

A.3.4.2 Unsupported fields

- snc_id(NULL).

A.3.4.3 Possible values for Supported fields

ne_id	This field will be mapped to nes.idnt_ov_objt field of OSMF database
ne_type	90
ne_name	This field will be mapped to nes.Selection_name field of OSMF database

COC of INM

counts_accurate	<ul style="list-style-type: none"> •xdr_accurate if nes.managed = 'y' and nes.available = 'y' and nes.aligned = 'y' •xdr_possibly_inaccurate If any of the three fields is not equal to 'y'
ne_tx_rate	xdr_ne_tx_rate_null(6)
ne_shelf_type	xdr_ne_shelf_ssu(23)
ne_support_bits	0
ne_counts	0
protection_type	xdr_protection_not_applicable(3)
protection_summary	0
valid_pm_stats	0

A.3.5 xdr_uoccoc_keep_alive_t
A.3.5.1 Supported fields

Nil

A.3.5.2 Unsupported fields

Nil

A.3.5.3 Possible values for Supported fields

Nil

A.3.6 xdr_uoccoc_ne_counts_inaccurate_t
A.3.6.1 Supported fields

- ne_id

A.3.6.2 Unsupported fields

Nil

Messages implemented between mainProcess(UOC) of OSMF-MOA and COC of INM

A.3.6.3 Possible values for Supported fields

ne_id	The field is mapped to the nes.idnt_ov_objt of OSMF database
-------	--

A.3.7 xdr_uoccoc_open_moc_session_t

A.3.7.1 Supported fields

- moc_major_sw_version
- moc_minor_sw_version
- moc_ip_address
- moc_name
- moc_function

A.3.7.2 Unsupported fields

- moc_my_serial(NULL)
- moc_peer_serial(NULL)

A.3.7.3 Possible values for supported fields

moc_major_sw_version	7
moc_minor_sw_version	0
moc_ip_address	IP adress of the OSMF-MOA platform.
moc_name	Hostname of the OSMF-MOA platform.
moc_function	xdr_primary

A.3.8 xdr_uoccoc_raise_alarm_t

A.3.8.1 Supported fields

- time
- ne_id
- alarm_id
- severity

Messages implemented between mainProcess(UOC) of OSMF-MOA‡**COC of INM**

- impact
- type
- unit_id
- reason_text
- threshold_crossing
- threshold_setting
- shelf_posn
- slot

A.3.8.2 Unsupported fields

- provision_change(FALSE)
- frame_id(NULL)
- frame_name(NULL)
- shelf_id(NULL)
- clifi(NULL)
- secondary_cpg(NULL)
- subslot(length = 0)
- problemType(NULL)
- probableCause(NULL)
- objectInstance(NULL)
- objectClass(NULL)
- trendIndication(NULL)
- specificProblem(NULL)
- threshold_count(Zero)

A.3.8.3 Possible values for Supported fields

time	This field will be mapped to The current_alarms.event_time of OSMF database.
ne_id	This field is mapped to nes.idnt_ov_objt of OSMF database
alarm_id	This field is mapped to Current_alarms.event# of OSMF Database
severity	This field is mapped to Current_alarms.severity of OSMF database

**Messages implemented between mainProcess(UOC) of OSMF-
MOA‡ COC of INM**

impact	<ul style="list-style-type: none"> •xdr_impact_service_affecting if current_alarms.sa = "SA" •xdr_impact_non_service_ <p>affecting if current_alarms.sa = "NSA"</p>
type	<ul style="list-style-type: none"> •xdr_alarm_type_equipment •xdr_alarm_type_service •xdr_alarm_type_environmental •xdr_alarm_type_unknown <p>The mapping between the SSU event types and the INM event types is provided in the section in this appendix entitled EVENT TYPE MAPPING</p>
unit_id	<ul style="list-style-type: none"> •"OUTA" •"OUTB" •"OUTC" •"OUTD" •"OUTE" •"ITH"
reason_text	This field is mapped to current_alarms.event_mesg of OSMF database
threshold_setting	String of the exceed performance parameter if it is a threshold event.
threshold_crossing	True if it is a threshold event. False if it is not a threshold event.
shelf_posn	<ul style="list-style-type: none"> •"S0" •"S1" •"S2" •"S3" •"S4"

**Messages implemented between mainProcess(UOC) of OSMF-MOA†
COC of INM**

slot	<ul style="list-style-type: none"> •1 •2 •3 •4 •5 •6 •7 •8 •9 •10
------	---

A.3.9 xdr_uoccoc_send_ne_list_t

A.3.9.1 Supported fields

- ne_id
- ne_type
- ne_name
- counts_accurate
- ne_tx_rate
- ne_shelf_type
- ne_support_bits

A.3.9.2 Unsupported fields

Same as for xdr_uoccoc_enroll_ne_t

A.3.9.3 Possible values for supported fields

Same as for xdr_uoccoc_enroll_ne_t

A.3.10 xdr_uoccoc_synchronize_t

A.3.10.1 Supported fields

Nil

A.3.10.2 Unsupported fields

Nil

A.3.10.3 Possible values for Supported fields

Nil

A.3.11 xdr_uoccoc_update_copc_info_t

A.3.11.1 Supported fields

Messages implemented between mainProcess(UOC) of OSMF-MOA and COC of INM

- copc_name
- copc_ip_address
- coc_tcp_port_number

A.3.11.2 Unsupported fields

Nil.

A.3.11.3 Possible values for supported fields

copc_name	The name of an already connected INM.
copc_ip_address	The IP address of an already connected INM.
coc_tcp_port_number	The port number of an already connected INM.

A.3.12 *xdr_uoccoc_update_mopc_activity_t*

A.3.12.1 Supported fields

- mopc_activity
- moc_name
- moc_function

A.3.12.2 Unsupported fields

- moc_my_serial(NULL)
- moc_peer_serial(NULL)
- moc_bridge_release(NULL)

A.3.12.3 Possible values for Supported fields

mopc_activity	•xdr_active
moc_name	Hostname of the OSMF-MOA Platform
moc_function	•xdr_primary

**Messages implemented between mainProcess(UOC) of OSMF-MOA†
COC of INM**

A.3.13 xdr_uoccoc_update_ne_counts_t

A.3.13.1 Supported fields

- ne_id

A.3.13.2 Unsupported fields

- ne_counts(Zero)

A.3.13.3 Possible values for Supported fields

ne_id	This field is mapped to nes.idnt_ov_objt of OSMF database
-------	---

A.3.14 xdr_uoccoc_update_mopc_support_t

A.3.14.1 Supported fields

- mopc_support_bits

A.3.14.2 Unsupported fields

Nil

A.3.14.3 Possible values for Supported fields

mopc_support_bits	1
-------------------	---

Messages supported between GNCE of INM† REGB of OSMF-MOA

A.4 Messages supported between GNCE of INM→ REGB of OSMF-MOA

- xdr_gnceregb_open_gnce_session_t
- xdr_gnceregb_request_mopc_info_t
- xdr_gnceregb_request_ne_type_info_t
- xdr_gnceregb_keep_alive_t

A.5 Messages supported between COC of INM→ REGD of OSMF-MOA

xdr_cocregd_open_coc_session_t
xdr_cocregd_register_copc_t
xdr_cocregd_unregister_copc_t
xdr_cocregd_keep_alive_t

**Messages supported between COC of INM→ mainProcess(UOC)
of OSMF-MOA**

**A.6 Messages supported between COC of INM→ mainProcess(UOC) of
OSMF-MOA**

xdr_cocuoc_accept_moc_session_t
xdr_cocuoc_keep_alive_t
xdr_cocuoc_request_alarms_t
xdr_cocuoc_request_ne_list

Messages supported between COC of INM and mainProcess(UOC) of OSMF-MOA

EVENT TYPE MAPPING

Keyword	Description	Enumerated value for INM
CFGCHD	Configuration changed	xdr_alarm_type_environmental
CFGRST	Configuration reset	xdr_alarm_type_environmental
INDISQ	Input disqualified	xdr_alarm_type_environmental
INEXPH	Input excessive phase hits	xdr_alarm_type_environmental
INMEAS	Disqualified by meas.	xdr_alarm_type_environmental
INPAIS	Input AIS	xdr_alarm_type_environmental
INPLOS	Input loss of signal	xdr_alarm_type_environmental
INPOOF	Input loss of framing	xdr_alarm_type_environmental
INPRCDQ	Input PRC status disqualified	xdr_alarm_type_environmental
INPULL	Pull-in range exceeded	xdr_alarm_type_environmental
MEASRNG	Measurement range exceed	xdr_alarm_type_environmental
MEASRP	Measurement beyond threshold	xdr_alarm_type_environmental
SNDRIFT	Drift exceeds limits	xdr_alarm_type_environmental
SNHOLD	Holdover mode	xdr_alarm_type_environmental
SNTRCK	Tracking this input	xdr_alarm_type_environmental
ITHACTV	ITH Active	xdr_alarm_type_equipment
ITHDISA	ITH disagree on channel quality	xdr_alarm_type_equipment
ITHFAIL	ITH failed	xdr_alarm_type_equipment
ITHHLTH	ITH Marginal failure	xdr_alarm_type_equipment

**Messages supported between COC of INM \ddagger mainProcess(UOC)
of OSMF-MOA**

ITHNCOM	ITH Communications failure	xdr_alarm_type_equipment
ITHNVFL	NVRAM failed	xdr_alarm_type_equipment
ITHOFFL	ITH Offline	xdr_alarm_type_equipment
ITHWARM	ITH Warmup	xdr_alarm_type_equipment
SNFREE	Free run mode	xdr_alarm_type_environmental
SNHAVL	Holdover available	xdr_alarm_type_environmental
SNOPER	Sync in normal Operating mode	xdr_alarm_type_environmental
SNSAVL	SmartClock available	xdr_alarm_type_environmental
INPSRM	Input SSM changed	xdr_alarm_type_environmental
SNSSM	SSM algorithm executed	xdr_alarm_type_environmental
ITHCDIN	Other ITH card Inserted	xdr_alarm_type_equipment
ITHCDRM	Other ITH card removed	xdr_alarm_type_equipment
LOGCLRD	Event log cleared	xdr_alarm_type_environmental
TRCLOS	TRSC Input Loss of Signal	xdr_alarm_type_environmental
TRCOOF	TRSC Input Out of frame	xdr_alarm_type_environmental
TRCSILP	TRSC Input Slip	xdr_alarm_type_environmental
TRCHSLP	TRSC Input High Slip	xdr_alarm_type_environmental
TRCBLOS	TRSC Buff Input Loss of signal	xdr_alarm_type_environmental

**Messages supported between COC of INM‡ mainProcess(UOC) of
OSMF-MOA**

OPCDFL	Output card failed	xdr_alarm_type_equipment
OPCDIN	Output card inserted	xdr_alarm_type_equipment
OPCDMS	Output card mismatch	xdr_alarm_type_equipment
OPCDNA	Output card not accessible	xdr_alarm_type_equipment
OPCDNP	Output card not present	xdr_alarm_type_equipment
OPCDPPM	Output card protection mismatch	xdr_alarm_type_equipment
IMCCRST	IMC Configuration reset	xdr_alarm_type_equipment
ITHGONE	ITH card missing	xdr_alarm_type_equipment
SNPWRA	Power supply A failed	xdr_alarm_type_service
SNPWRB	Power supply B failed	xdr_alarm_type_service
SWDL	Software download mode	xdr_alarm_type_environmental
SNFAIL	Sync failure	xdr_alarm_type_service
TDEV	___	xdr_alarm_type_environmental
MRTIE	___	xdr_alarm_type_environmental
LMRTIE	___	xdr_alarm_type_environmental
FFOFF	___	xdr_alarm_type_environmental
SPRD	___	xdr_alarm_type_environmental