# Oracle® Communication and Mobility Server

Installation Guide

10g Release 3 (10.1.3)

**E12657-02**

July 2008

ORACLE®

Oracle Communication and Mobility Server Installation Guide 10g Release 3 (10.1.3)

E12657-02

# Contents

## 3 Verify the OCMS Installation and Features



## 4 Install Oracle Communication and Mobility Server with a Backend Oracle RAC Database



## 5 Presence Large Deployment Installation

# Preface

This preface contains the following topics:



## Audience

This guide is intended for system administrators and developers who want to install and use Oracle Communication and Mobility Server.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## Related Documents

For more information, see the following documents in the Oracle Communication and Mobility Server, Oracle Containers for J2EE, Oracle Application Server, and Oracle TimesTen In-Memory Database product sets:

- *Oracle Communication and Mobility Server Administrator's Guide*
- *Oracle Containers for J2EE Configuration and Administration Guide*
- *Oracle Containers for J2EE Deployment Guide*
- *Oracle Application Server Installation Guide*
- *Oracle Application Server Administrator's Guide*
- *Oracle Communication and Mobility Server Release Notes*

## Conventions

The following text conventions are used in this document:

| Convention | Meaning |
|---|---|
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

# 1

# Product and Installation Overview

This chapter describes Oracle Communication and Mobility Server (OCMS) and recommended topologies in the following sections:



## Product Overview

OCMS is a carrier grade SIP (Session Initiation Protocol) and J2EE execution platform for applications needing to send or receive messages over a SIP enabled broadband network or cellular network. This platform is referred to as a SIP Application Server.

Examples of such applications are as follows:

- IP Telephony
- Video Calling
- Instant Messaging
- Speed Dial Service
- Call Forwarding Service
- Third Party Call Control
- Emergency Call Service

OCMS can be deployed either for broadband or 3GPP IP Multimedia System (IMS) networks offering value-added services.

### New in this Release

This release of Oracle Communication and Mobility Server includes enhancements and new features. To read about new and improved features, see *Oracle Communication and Mobility Server Administrator's Guide*, and *Oracle Technology Network* at http://www.oracle.com/technology/products/ocms/otn_front.htm.

## Installation Prerequisites

This section describes prerequisites for installing OCMS and for using specific OCMS functionality.

## System Requirements

To install OCMS your system must meet the following requirements:

- Red Hat Enterprise Linux AS 4 (update 5 or higher), AS 5 (update 1 or higher)
- Recommended for development purposes only: Microsoft Windows 2000 with Service Pack 3 or above, Microsoft Windows Server 2003 (32-bit) with Service Pack 1 or above, or Microsoft Windows XP Professional with Service Pack 2
- Java 2 Platform, Enterprise Edition Software Development Kit (JDK) 5.0, (JDK version 1.5). Oracle Application Server installation mode uses the JDK 1.5.0 from the installation of Oracle Application Server. Standalone developer mode bundles JDK 1.5.0 with the OCMS installation.
- Oracle 10g Database (10.2.0.3 or higher) or Oracle Database 11g (11.1.0.6 or higher), for Subscriber Data Services and Presence.
- Refer to What Components are Installed for more information on Subscriber Data Services.
- a LAN connection and an IP address and hostname

**Note:** The listed operating systems are certified and supported for Oracle Communication and Mobility Server. Other operating systems have not been certified.

## Port Requirements

OCMS requires the use of ports for TCP and UDP communications for the SIP Container, Presence server, and, if installed, the Edge Proxy. Default port numbers will be specified for the SIP Container and Edge Proxy ports, or use port numbers that you specify. The Presence port is not configurable during installation, but can be configured in the MBean browser. Refer to the *Oracle Communication and Mobility Server Administrator Guide* for information on how to reconfigure port numbers following installation.

During installation, selected ports are verified. If a port is in use by another application, the installation will not succeed, and an error message will appear.

Table 1–1 lists the default ports used by OCMS when you install the Edge Proxy application.

***Table 1–1 Default Ports: Edge Proxy Installed***

| SIP Port | Edge Proxy Port | Presence Port |
|---|---|---|
| 5070 | 5060 | 5081 |

Table 1–2 lists the default ports used by OCMS when you do not install the Edge Proxy application:

***Table 1–2 Default Ports: Edge Proxy Not Installed***

| SIP Port | Edge Proxy Port | Presence Port |
|---|---|---|
| 5060 | N/A | 5070 |

### Checking if a Port is in Use

To check if a port is being used in Windows, you can run the netstat command as follows:

C:\> **netstat -an | find** "*portnum*"

Note that you need double quotes around the port number.

To check if a port is being used in Linux, you can run the netstat command as follows:

**netstat -a | grep** <portnum>

## Installation Modes

Oracle Communication and Mobility Server can be installed in the following installation modes:

- Install Standalone Developer Mode
- Install to Existing Oracle Application Server

Table 1–3 lists requirements for each installation mode.

***Table 1–3 Requirements for Installation Modes***

| Requirement | Oracle Application Server mode | Standalone Developer mode |
|---|---|---|
| Oracle Containers for J2EE (OC4J) | Required before installation | Not required. Included with the installation. |
| Java Development Kit (JDK) 1.5 | Provided by Oracle Application Server. | Included. |

### Install Standalone Developer Mode

The standalone developer mode allows developers to develop and test SIP applications in a minimal installation environment, without an application server. This installation mode installs Oracle Containers for J2EE (OC4J).

Standalone developer mode is not typically recommended for production deployment because of limited management capabilities.

### Install to Existing Oracle Application Server

The recommended installation mode installs OCMS into an Oracle Application Server Release 10.1.3.4 or higher environment. This installation mode allows OCMS to use OC4J features such as Oracle High Availability (HA), clustering, and replication.

Oracle Communication and Mobility Server is deployed with an OC4J container that you manage using the Oracle 10g Enterprise Manager Application Server Control console. The Application Server Control functions enable starting, stopping, restarting, deploying, undeploying, and redeploying applications

Additionally, the Application Server Control MBean browser enables you to configure and manage the OCMS components. Configuring the attributes of the OCMS MBeans (Managed Beans) enables you to execute such administrative tasks as configuring the DNS (Domain Name System), configuring and managing Presence, and the basic configuration (port, IP, and host address) of the OCMS SIP Server itself.

For more information on managing and configuring OCMS in an Oracle Application Server environment, refer to "Managing the SIP Server" in the *Oracle Communication and Mobility Server Administrator's Guide*.

For more information on features provided by Oracle Application Server, refer to *Oracle Application Server Administrator's Guide*.

## What Components are Installed

The following applications are selected for installation by default, unless the user de-selects them during installation.

- **SIP Container**: A SIP Servlet Container extends the J2EE Application Server, providing a runtime environment for SIP applications, including services such as security, concurrency, lifecycle management, transaction, deployment, and other services. A JSR116-compliant SIP Servlet Container provides network services for sending and receiving SIP requests and responses using a combination of transport protocols, IP addresses, and port numbers to listen for incoming SIP traffic. The OCMS SIP Servlet Container can be installed on an existing instance of Oracle Application Server, running in OC4J. Alternatively, the OCMS SIP Servlet Container can run on its own standalone instance of OC4J. The typical OCMS SIP Servlet Container is composed of an Oracle Application Server instance with OC4J as its J2EE container, and Oracle Process Manager and Notification Server (OPMN) to monitor the server. OCMS currently supports high availability deployments in this configuration only.
- **Subscriber Data Services**: Subscriber Data Services is installed when applications require the authentication and security services provided by OCMS. Subscriber Data Services requires Oracle Database.
- **Proxy Registrar**: Proxy Registrar combines the functions of a SIP Proxy Server and Registrar. Its main tasks include registering subscribers, looking up subscriber locations, and proxying requests to destinations.

  The Proxy Registrar depends on the Subscriber Data Services for the Location Service and the registrar subcomponents of the Proxy Registrar to function.
- **Presence**: The Presence application enables a service provider to provide Presence services to end users as well as enabling the service provider to base other services on Presence information (for example, intelligent call routing based on Presence information).

  The Presence application depends on the SIP Container.
- **Aggregation Proxy**: Aggregation Proxy authenticates any XCAP requests and Web Services calls. The XCAP requests and Web Services calls are then proxied to their respective servers. For more information, refer to *Oracle Communication and Mobility Server Administrator's Guide*.

  The Aggregation Proxy depends on the SIP Container and Subscriber Data Services.
- **Application Router**: Application Router is a SIP application that routes incoming SIP requests to the correct application. It is a required component for composing applications or for deploying new applications of your own.

  The Application Router routes requests by placing route headers in each SIP request it processes. A number of route headers can be placed in a request, each representing a different destination URI. The SIP request is either sent through the

chain of destination URIs, or proxied to a new URI after arriving at its first destination.

The Application Router application depends on the SIP Container and Subscriber Data Services.

The following applications are not installed by default, and must be selected for installation by the user from the list of available applications:

- **Parlay X Presence Web Services**: Presence Web Services provides support for the Parlay X Presence Web Services. This is an implementation of the Parlay X 2.1 Web Services interface.

  The Presence Web Service application depends on the SIP Container

- **STUN Server**: The STUN server implements the STUN (Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators) server protocol. This acts as a NAT traversal mechanism to allow clients to find out the external IP address and port that the client is represented by.

  The STUN server depends on the SIP Container.

- **Edge Proxy**: Edge Proxy provides a SIP distribution for routing requests, proxying SIP requests to a particular OCMS SIP Server instance. The Edge Proxy forms logical pathways between individual clients and SIP servers, such that SIP traffic sent from a particular client session is always handled by the same server. As the number of SIP clients increases, additional Edge Proxy servers can be added, providing highly scalable handling of SIP clients.

  The Edge Proxy is available for installation only in Oracle Application Server Mode.

- **User Dispatcher**: The User Dispatcher enables stateful applications, such as the Presence applications, to scale. The User Dispatcher is a proxy that dispatches SIP and XCAP (over HTTP) requests to the their appropriate destinations on a consistent basis.

For more description and configuration information on these features, refer to *Oracle Communication and Mobility Server Administrator's Guide*.

# 2

# Install Oracle Communication and Mobility Server

This chapter describes how to perform an installation of Oracle Communication and Mobility Server (OCMS). It contains the following sections:



## Install a Cluster of OCMS Instances in a High Availability Environment

You can install OCMS either as a single node for evaluation or development, or you can install OCMS to multiple nodes for a high availability production topology. For a high availability production topology you will need one or more Edge Proxies nodes along with the other OCMS nodes. Edge Proxies provide scalability and high availability. Edge Proxies are required for advanced production level topologies, and are usually used to provide SIP distribution. The Edge Proxy distributes incoming SIP traffic among OCMS SIP application servers when used between a SIP unaware load balancer and an OCMS cluster.

For information on recommended deployment topologies and configuring high availability see the "Deployment Topologies" and "Configuring High Availability" chapters in the *Oracle Communication and Mobility Server Administrator's Guide*.

### Configure OCMS in a Clustered Environment with Edge Proxies

An administrator typically installs Edge Proxies on separate nodes. The installation of an Edge Proxy as part of an OCMS installation is supported only in Oracle Application Server installation mode. An Edge Proxy is aware of multiple OCMS instances through OPMN clustering. This requires a clustered Oracle Application Server 10.1.3.4 environment where each OCMS instance references a unique Oracle Home.

### Size the Installation

The number of Edge Proxies recommended for an OCMS installation depends on scalability and high availability requirements for the installation and on the number of SIP clients and OCMS instances. A minimum of two Edge Proxies is recommended to

guarantee high availability in a clustered environment. As the number of SIP clients or OCMS instances grows, additional Edge Proxy servers can be added. Refer to the *Oracle Communication and Mobility Server Administrator's Guide* for more information.

In OCMS installations with three or more OCMS instances, Edge Proxies are typically separately installed OCMS instances that consists of only the Edge Proxy application (that is, only Edge Proxy per OCMS instance is selected for installation). For OCMS installations with not more than two instances, Edge Proxies can be installed concurrently with the OCMS instances.

## Assign Ports in a Multi-Instance Installation

After you have decided on the number of OCMS instances and Edge Proxy-only instances in the OCMS installation, you must decide what ports will be assigned to each instance. For single machine installations, each instance in the OCMS installation must have uniquely defined ports. The OCMS installer verifies that ports are available at the time of installation. Example 2–1 shows how to assign ports in a multi-instance OCMS installation with an Edge Proxy server.

***Example 2–1 Assigning Ports in a Multi-Instance OCMS Installation with an Edge Proxy***

```
OCMS instance 1: Custom installation consisting of Edge Proxy only
Edge Proxy Port: 5060 (default)

OCMS instance 2: Typical installation
SIP Port: 5080 (user-configured during installation)

OCMS instance 3: Typical installation
SIP Port: 5090 (user-configured during installation)

OCMS instance 4: Typical installation
SIP Port: 5100 (user-configured during installation)
```

## Install Edge Proxies

Perform the following steps to install Edge Proxies into an OCMS installation:

1. In the **Available Product Components** screen of Oracle Universal Installer, select the **Edge Proxy** component.

   **Note:** No other component should be selected, only the Edge Proxy.

2. Click **Next**.
3. Complete the OCMS installation.
4. Perform an **opmnctl status** command to see the status of the Edge Proxy.

   The Edge Proxy runs automatically after being installed and reports a status of "Alive."

5. Run the Oracle Universal Installer and repeat the steps above to configure additional Edge Proxies.
6. Perform the following additional procedures in the *Oracle Communication and Mobility Server Administrator's Guide*:
   - "Configuring the OCMS SIP Containers for High Availability"
   - "Configuring the Edge Proxy Nodes for High Availability"

- "Configuring Highly Available SIP Servlet Applications"

7. Install the non-Edge Proxy OCMS instances using the procedures in this guide.

   After installation of all the Edge Proxies and OCMS instances you will be able to view the topology of the OCMS installation through Enterprise Manager from the Oracle Application Server instance where "Start AS Control" was selected during Oracle Application Server installation.

## Install Oracle Communication and Mobility Server

You can install OCMS on the Microsoft Windows or Linux operating systems.

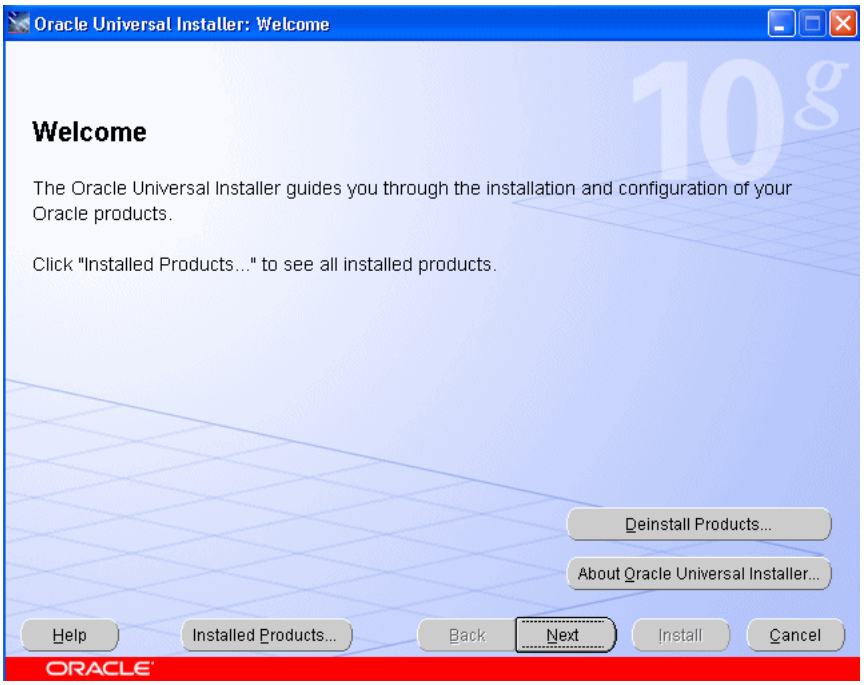
Perform the following steps to install OCMS:

1. Close any SIP client applications that are running.
2. Run the installation executable for your operating system:
   - For Microsoft Windows operating systems, run **setup.exe**.
   - For Linux operating systems, run **runInstaller**.

   Oracle Universal Installer starts and automatically installs Java 2 Platform, Standard Edition Development Kit (JDK) 1.5.0.0.6.

   The Welcome screen appears.

***Figure 2–1 Welcome screen***




**Note:** Each screen in Oracle Universal Installer contains a **Help** button. Use the Help to get more information about the screen.

3. Click **Next** to continue. The Specify File Locations screen is displayed.

4. Ideally, you should not modify the stage location. When installing into an existing Oracle Application Server installation, ensure that you select the existing ORACLE_HOME as the installation location. Do not leave any fields blank. Click **Next** to continue.

## Select an Installation Type

This screen enables you to select the Installation Type.

1. Select the Installation Type you want t perform:
   - *Install in standalone developer mode*: The standalone developer mode is recommended for development users only due to limited management capabilities. This option installs Oracle Containers for J2EE (OC4J) software, Release 10.1.3.4, and does not require an application server.
   - *Install to existing Oracle Application Server (release 10.1.3.4 is supported)*: This installation mode is the recommended selection for production deployments. Oracle Universal Installer verifies if the provided Application Server Home contains an Oracle Application Server installation.

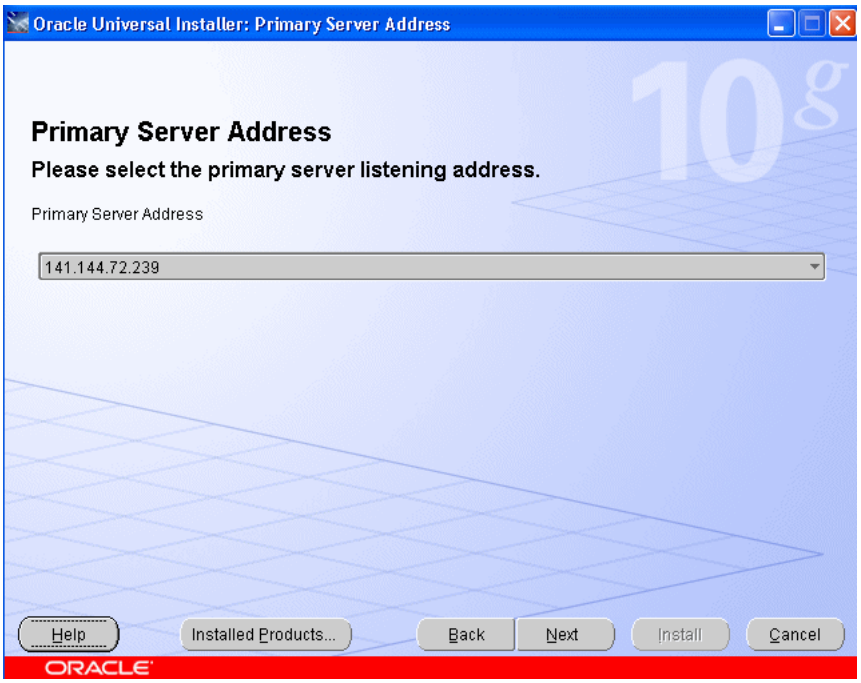
     **Note:** When installing into an existing Oracle Application Server, Oracle Universal Installer automatically names the OC4J instance *ocms*. Before installing OCMS, ensure that you do not have any other instances of OC4J named *ocms* in your ORACLE_HOME.

2. Click **Next** to continue. The Available Product Components screen is displayed.

## Specify the Primary Server Address

The Primary Server Address is the address that clients will use to communicate with OCMS.

*Figure 2–2 Primary Server Address screen*




1. Select the primary server listening address from the drop-down menu.
2. Click **Next** to proceed. The OC4J Admin and Password screen appears.

## Specify the OC4J Admin and Password

If installing in standalone mode, the password provides for validation in stopping the OCMS instance.

If installing in Oracle Application Server mode, entering the administrator password allows for immediate validation of these credentials by the Oracle Application server.

*Figure 2–3 OC4J Admin screen*


1. Enter the Administrator username.
2. Enter the Administrator password.
3. Click **Next** to proceed. The Oracle DB Details screen appears.

---

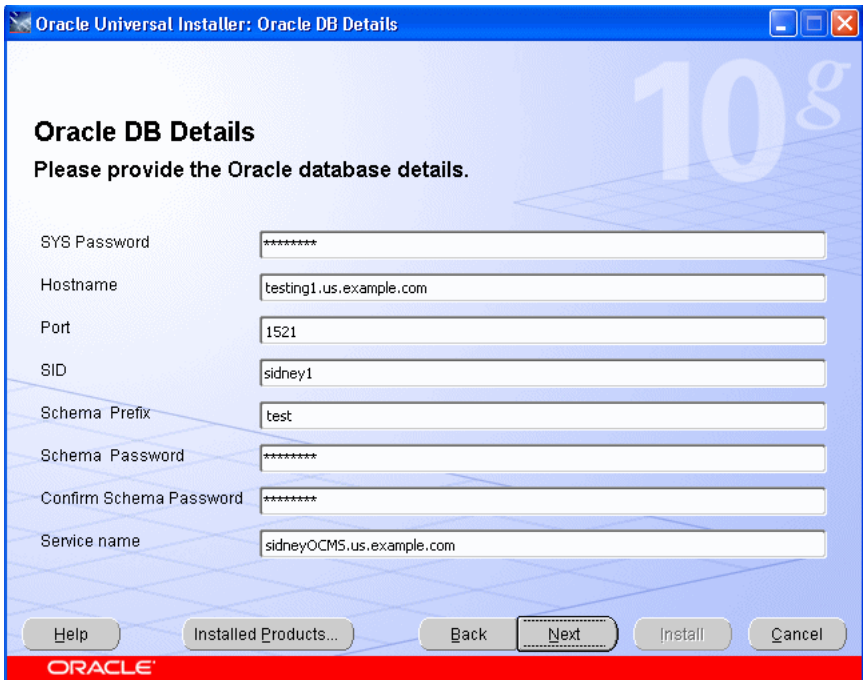
**Note:** After these initial screens, the order of presentation of screens for the installation will vary depending upon on your product/component selections. The following screens are presented in a certain order, but that presentation is not meant to imply that your installation will proceed in the same order.

---

## Provide Oracle DB Details

Use this screen to enter the details about your Oracle database.

*Figure 2–4*

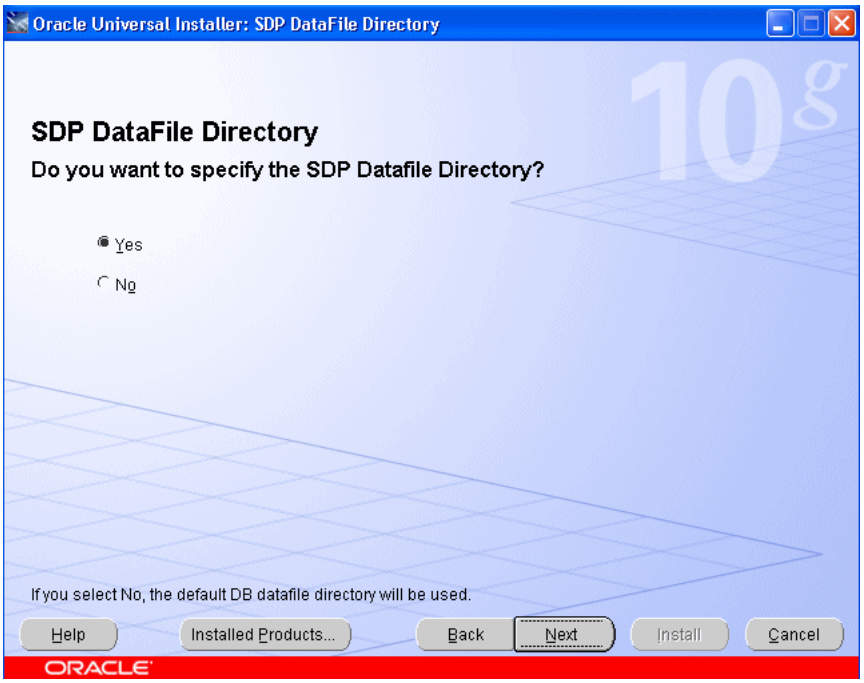



1. Fill-in the following fields (your database credentials will be validated when you connect to the database):
   - SYS Password
   - Hostname
   - Port
   - SID
   - Schema Prefix
   - Schema Password (and confirmation of same)
   - Service Name
2. Click **Next** to proceed.

## SDP Datafile Directory

Use this screen to specify whether or not you want to specify the SDP Datafile directory.

*Figure 2–5 SDP Datafile Directory screen*




- If you choose to specify the directory, the next screen (SDP Datafile Location) enables you to input that information.

  **Note:** You must ensure that the database data file directory is present and writable. If it is not, then the configuration tool will fail.

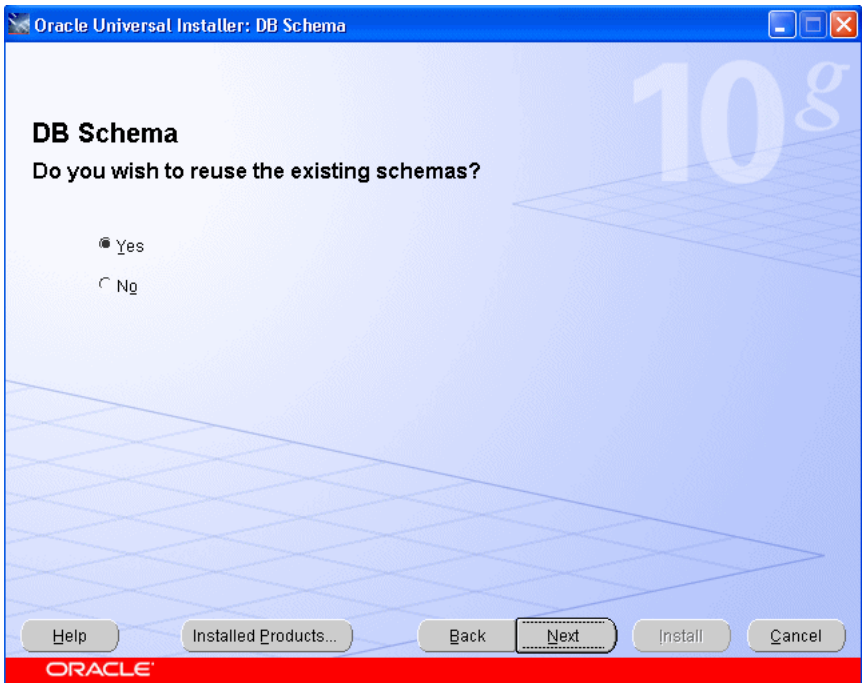
- If you do not want to specify a location, the default DB datafile directory will be used.

Click **Next** to proceed.

## DB Schema

Select **Yes** or **No** to reuse the existing schema, then click **Next** to proceed. The STUN Server Configuration screen appears, if you chose to install a STUN Server earlier.

*Figure 2–6 DB Schema screen*




## Specify the STUN Server Configuration

If you selected STUN Server for installation, you will be prompted to enter the host and port for the primary and secondary STUN servers. Configuring both primary and secondary STUN servers is required.

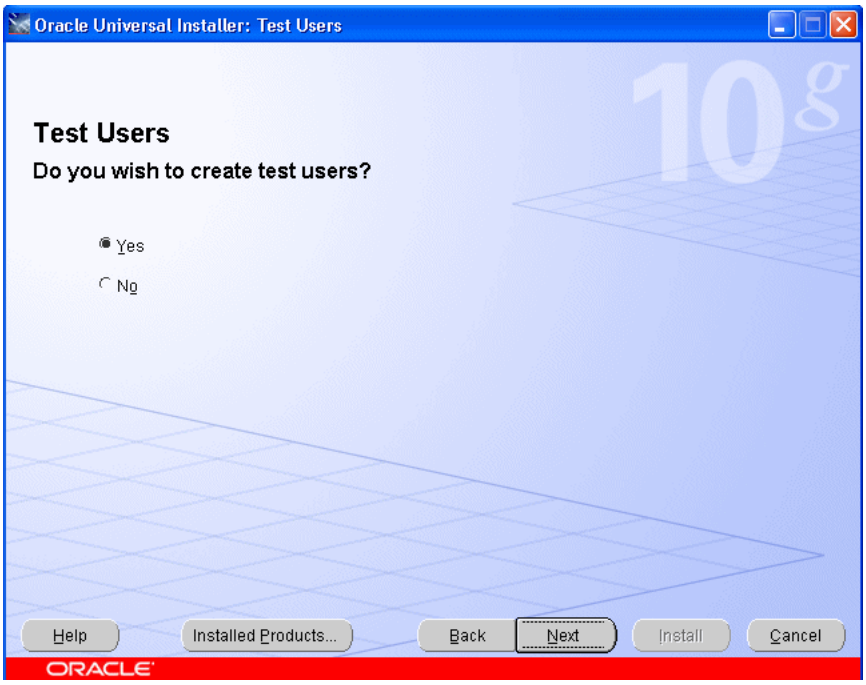
To specify the STUN Server hosts and ports:

1. Enter the hostname or IP address of the primary STUN server in the *Primary Host Address* field.
2. Enter the Primary Port.
3. Enter the hostname or IP address of the secondary STUN server in the *Secondary Host Address* field.
4. Enter the Secondary Port.
5. Click **Next** to continue.

## Configure Test Users

When installing OCMS you can create a number of predefined test users. To create test users, click **Yes**, then **OK** to proceed. The Test User Details screen appears.

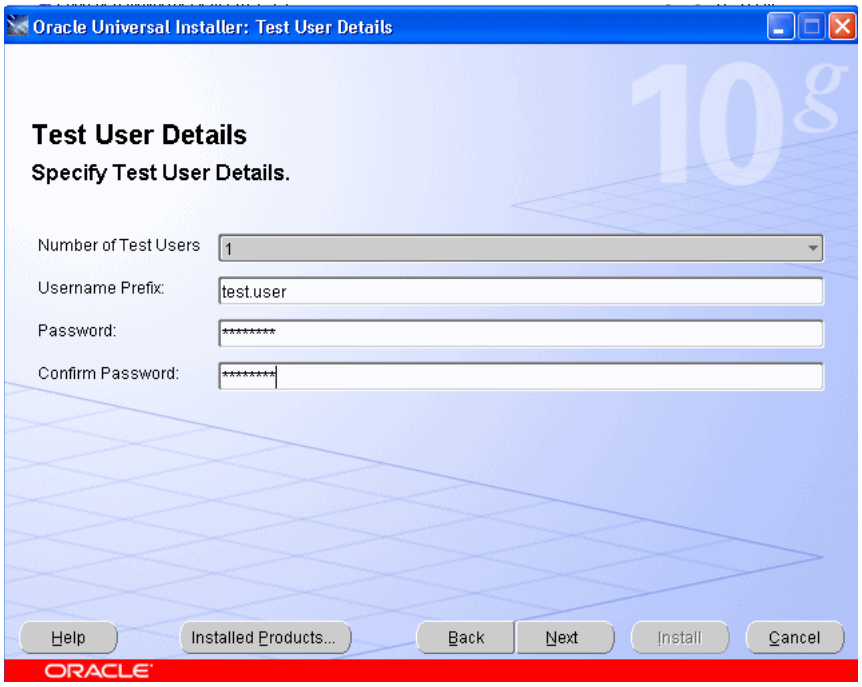
*Figure 2–7 Test Users screen*




## Test User Details

Use this screen to enter details about test users.

*Figure 2–8 Test Users Details screen*




1. Select the **Number of Test Users** from 1-12.

2. Enter the **Username Prefix** for the test users.

   The default prefix is "test.user". The username for each test user will be prefixed by this string.

   For example, if you select 3 for Number of Test Users and "test.user" for Username Prefix, the following test users will be created:

   ```
   test.user1@sip-domain
   test.user2@sip-domain
   test.user3@sip-domain
   ```

   **WARNING: This release of OCMS supports lowercase usernames. Only use lowercase usernames for the Username Prefix.**

3. Enter the **Password** for the test users.

   Each test user will have this password.

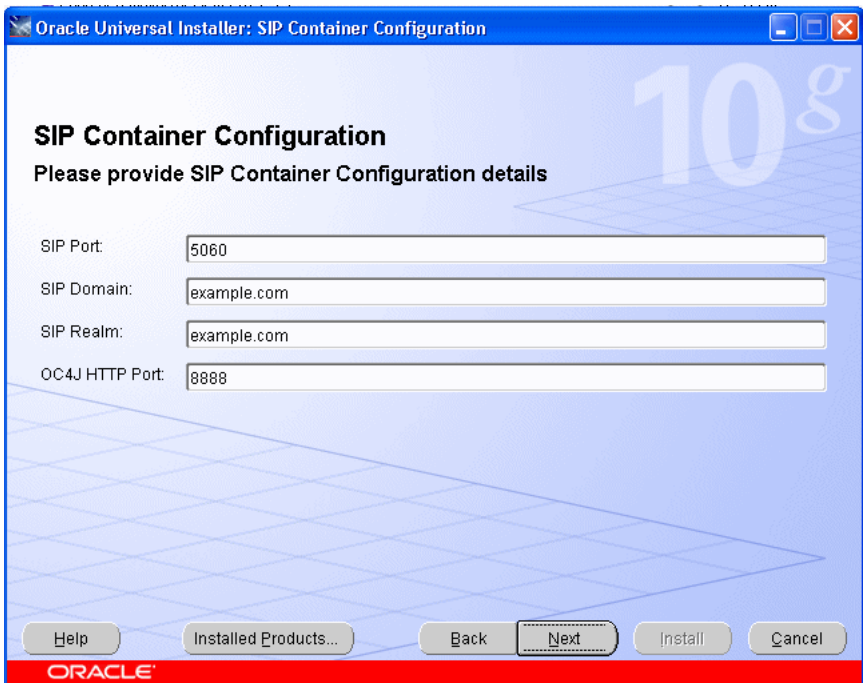
4. Confirm the **Password** for the test users.

5. Click **Next**.

## Configure the SIP Container

Provide the following SIP Container information:

***Figure 2–9 Configure SIP screen***




- **SIP Port**: The default SIP port is displayed. The port used is 5070 in an installation with Edge Proxy, and is port 5060 in an installation without Edge Proxy.

  If you receive a message that the SIP port is in use, exit all client applications, select Back, and return to this window.

- **SIP Domain**: Specifies the domain or hostname of the machine where OCMS is being installed. The default is example.com.
- **SIP Realm**: Specifies the SIP realm used for authentication. This is also the domain or hostname of the machine where OCMS is being installed. The default is example.com.
- **OC4J HTTP Port** (standalone developer mode only): Specifies the HTTP port used to manage OCMS through OC4J in standalone developer mode. The default port is 8888. If another application uses this port you can select another port.

  In Oracle Application Server mode you will not have the option to set the OC4J HTTP Port as it is autodetected by the installer from your Oracle Application Server installation. For installation to an Oracle Application Server which does not have the HTTP_Server component installed, the installer will automatically pick the first free port in the range *7785 – 7799*.

---

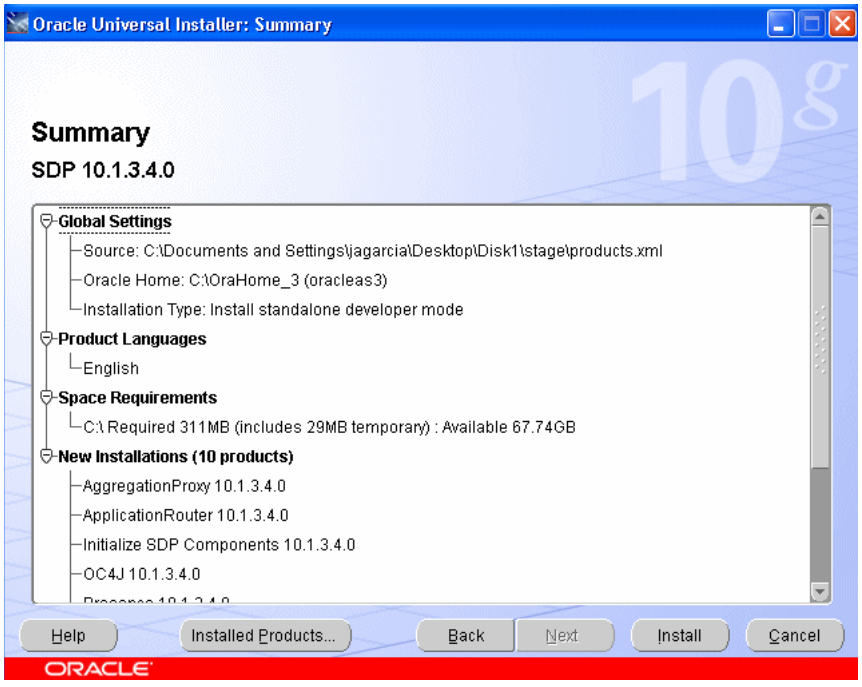
**Note:** Oracle Universal Installer enables configuration of only one SIP domain and one SIP realm. You can configure additional SIP domains and SIP realms after the installation using the Domains and Realms attribute of the SIP Servlet Container MBean. Multiple SIP domains can exist in a SIP realm. Refer to *Oracle Communication and Mobility Server Administrator's Guide* for more information.

---

## Summary Information

The OCMS installation summary screen displays the installation environment, the components to be installed, and the settings and port numbers that you configured.

***Figure 2–10 Summary screen***




1. Record the installation summary information in the "Installation Configuration" section, including port numbers, for your later reference.

This information is required when configuring the SIP client.

2. Ensure that all the installation summary information is correct, and the correct OCMS applications will be installed.
3. Click **Install** to begin installing the OCMS components. The End of Installation screen appears when all of your selected products/components have been installed.

> **Note:** If you encountered any errors during the installation, look at the installation log file at $HOME/oraInventory/logs

4. Click **Exit** to exit the installation.

## List the Oracle Remote Method Invocation (RMI) Port

Oracle Application Server Containers for J2EE (OC4J) provides support for allowing EJBs to invoke one another across OC4J containers using the proprietary Remote Method Invocation (RMI)/Oracle RMI (ORMI) protocol. For more information on Oracle RMI, refer to *Oracle Application Server Containers for J2EE Services Guide*.

If a particular JMX application requires connecting to the Oracle RMI port, then you will need to know what port is assigned for RMI. The RMI port is dynamically assigned by OPMN (Oracle Process Management and Notification).

The following command lists the latest port assignments:

```
opmnctl status -l
```

For example (some columns omitted for clarity):

***Figure 2–11 Output from opmnctl command***

```
Processes in Instance: <instancename>_<myhost>.com
-----------------+-------------------+-----+-----+-----
ias-component    | process-type      | pid |...| ports
-----------------+-------------------+-----+-----+-----
OC4JGroup:ocms   |OC4J:ocms          |7156 |...|jms:12603,ajp:12503,rmis:12703,sip:5060,rmi:12403
OC4JGroup:ocms   |OC4J:OC4J_WebCent| N/A |...|N/A
OC4JGroup:ocms   |OC4J:home          |7154 |...|jms:12601,ajp:12501,rmis:12701,rmi:12401
HTTP_Server      |HTTP_Server        |7153 |...|https1:4443,http2:7200,http1:7777
ASG              |ASG                | N/A |...|N/A
```

The ports column of the table lists the ports selected by opmn. The following example corresponds to the OCMS instance of OC4J ("OC4J: ocms"):

```
jms:12603,ajp:12503,rmis:12703,sip:5060,rmi:12403
```

## Verify the OCMS Installation

Perform the following steps to verify that OCMS has been successfully installed and is running:

1. With the OCMS server up and running, perform the following command to make sure that the SIP container is listening for SIP traffic on port 5060 (default SIP port):

   ```
   netstat -a | grep 5060
   ```

   Replace "5060" with the port number you chose during installation. You should see the following output:

```
tcp       0      0 <hostname>:5060  *:*       LISTEN
udp      0     0 <hostname>:5060  *:*
```

2. Navigate to the Oracle Enterprise Manager Web page at http://<ip_address>:<port_number>/em.

   For port number, use one of the following values:

   - Use 8888 for standalone developer mode installation.
   - Use the HTTP server port for an Oracle AS installation with an HTTP server.
   - Use 7785 for an Oracle AS installation without an HTTP server.
   - For HTTPS, use the HTTPS port for the Oracle AS installation.

3. Enter the OC4J administrator *username* and *password*.
4. Verify that Oracle Home specifies the OCMS installation directory.
5. Verify that server Status is "Up."
6. Select the **Applications** tab.
7. Verify that the applications you installed are deployed and are all up and running. You may notice that one or more applications are deployed as a child application of subscriberdataservices.

   For a typical installation, you should see ocmsrouteloaderear, proxyregistrar, subscriberdataservices, and presence.

8. Return to the home page and select the **Administration** tab.
9. Select **JMX/System MBean Browser**. In the left margin, expand SipContainer and click on SipServletContainer and verify the MBean property values

The verification of OCMS is complete.

## Start and Stop Oracle Communication and Mobility Server

After installation, you can start Oracle Communication and Mobility Server. You can manually start or stop OCMS as described in the following procedures.

To start OCMS in a Linux operating system, enter the following commands:

- cd $ORACLE_HOME/sdp/bin
  ./startocms.sh

To stop OCMS in a Linux operating system, enter the following commands:

- cd $ORACLE_HOME/sdp/bin
  ./stopocms.sh

To start OCMS in a Windows operating system, enter the following command:

- Run the startocms.bat file from the <*ocms_directory*>\sdp\bin directory.

To stop OCMS in a Windows operating system, enter the following command:

- Run the stopocms.bat file from the <*ocms_directory*>\sdp\bin directory.

For installation to an existing Application Server, you can also start OCMS by executing the following:

```
$ORACLE_HOME/bin/opmnctl startproc process-type=ocms
```

To stop OCMS in an existing Application Server:

```
$ORACLE_HOME/bin/opmnctl stopproc process-type=ocms
```

## Start and Stop Edge Proxy

You can start and stop the Edge Proxy in an existing Application Server using the following command:

```
$ORACLE_HOME/bin/opmnctl startproc process-type=EdgeProxy
```

To stop the Edge Proxy in an existing Application Server:

```
$ORACLE_HOME/bin/opmnctl stopproc process-type=ocms
```

## Deinstall

Deinstallation of individual products/components is not possible; you must deinstall OCMS (using Oracle Universal Installer) in order to remove products.

## Troubleshoot Installation Issues

This section contains information about solving installation-related problems that some users have encountered.

### Port Conflicts

As described in "Port Requirements" in the Product and Installation Overview chapter of this guide, OCMS requires the use of specific ports for SIP, Presence, and Edge Proxy communications.

To verify if a port is in use, following the instructions in Checking if a Port is in Use.

If there is a conflict, you can change a port number, or stop the other service that is using that port number.

### Loss of Network Connection During Installation

During the OCMS installation, an active network connection is required.

In the uncommon event that a network connection is lost during installation, the installation can fail. Examine the `installActions<data_time>.log` file (in the Oracle inventory directory) to see a log of this event.

---

**Note:** For more information about issues with the product, see the Release Notes that accompany this documentation.

---

# 3

# Verify the OCMS Installation and Features

This chapter describes how to verify the installation of Oracle Communication and Mobility Server (OCMS). It contains the following topics:

- Install Oracle Communicator and Verify the OCMS Installation

## Install Oracle Communicator and Verify the OCMS Installation

Before deploying an application, verify that you can connect to Oracle Communication and Mobility Server by provisioning two users, and installing a SIP client, Oracle Communicator, to perform a connectivity test. Oracle Communicators clients listen on the destination IP address and port configured in the deployment descriptor.

You can use Oracle Communicator to test the following:

- Interaction with the Presence Server—Subscribe to and publish a user's Presence, receive event notifications, for example watch for a popup indicating that a user has come online.
- Voice communication—Place a call from one user to another to verify VoIP functionality.
- Instant Messaging—Send a message from one user to another to verify the functionality of instant messaging.

Perform the following tasks to set up the Oracle Communicator client and test OCMS:

- Provision Sample Users—Each sample user is used to login to an instance of the Oracle Communicator client.
- Set the Log Level—Setting the log level to "info" to enable viewing all SIP traffic in real time.
- Install and Configuring Oracle Communicator—Installing two instances of the Oracle Communicator client in order to test the functionality of the core OCMS SIP servlets. Each instance should be installed and run on a different computer.
- Verify Servlet Registration—Verifying that the servlet has been registered correctly.
- Test the Presence Server—Testing publication and subscription to a user's Presence, and receipt of event notifications.
- Make a SIP Test Call—Establishing a call through the SIP network.
- Make a SIP to PSTN Test Call—Establishing a call from a test user to a telephone in the public switched telephone network (PSTN).

## Provision Sample Users

If you created test users during installation, you are ready to connect to your installation of OCMS using Oracle Communicator. You can install and configure Oracle Communicator and sign-in as one of the test users you created during installation.

Otherwise if you did not create any test users, you must first provision two users using SASH in order to test the SIP servlet by navigating to the following directory and launching SASH: `cd $ORACLE_HOME/sdp/sash/sbin`.

For information on provisioning users, refer to "Provisioning Users and Applications" in *Oracle Communication and Mobility Server Administrator's Guide*.

**WARNING: This release of OCMS supports lowercase usernames. Only use lowercase usernames.**

## Set the Log Level

Set the log level to `info` to log all traffic. Watch the log while registering the servlet and communicating with the clients. For more information, refer to "Setting the Log Levels for Core Components" in the *Oracle Communication and Mobility Server Administrator's Guide*.

Keep an open logger window so as to monitor the tests conducted in this chapter.

## Install and Configuring Oracle Communicator

If two different computers are available, you can install and configure two instances of the Oracle Communicator client on two different computers. If two computers are not available, you can start two instances of Oracle Communicator on the same computer. In that case, you must make sure that OCMS is running first, provision two accounts, then start two instances of Oracle Communicator on the same computer.

To install and configure the Oracle Communicator client:

1. Ensure that Oracle Communication and Mobility Server is running.
2. Launch the installation file and follow the on-screen instructions.
   - Click **Next** at the Welcome screen.
   - Select the installation directory for Oracle Communicator and click **Next**.
   - Select the installation preference and click **Next**.
   - Click **Next** to begin the installation.
   - Click **Finish** when the installation is complete.
3. Run the Oracle Communicator client by selecting **Start > Programs > Oracle > Oracle Communicator** and complete the Audio Setup wizard.

   The AudioSetup wizard allows you to configure and test your computer's audio hardware, and configure sound devices. Perform the following configuration:

   - Connect a headset to your computer and click **Next**.
   - Read the test paragraph aloud. To adjust the volume click the **Volume** button and make any necessary changes.

     The hardware wizard indicates whether or not your voice was detected.

- Click **Finish** to complete the audio configuration.

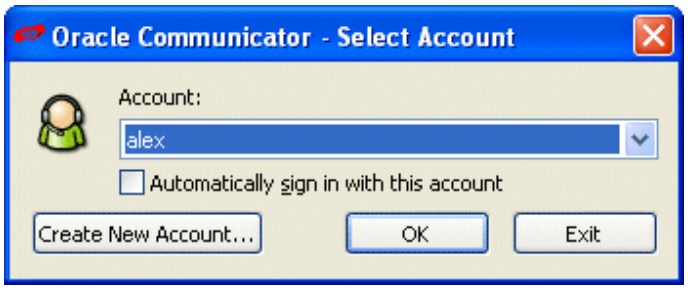
  The Oracle Communicator Create Account Wizard is displayed, prompting you to create a new account.

4. Enter a name for the account to be created and click **Next**. This account name must be one you provisioned using the Sash command line.
5. Enter a SIP address for the first user in the form `username@example.com`.
6. Enter the full name of the user.

   This is the name that is sent with each message.
7. Click **Finish** to exit the wizard.

   The Select Account window is displayed (Figure 3–1).

***Figure 3–1 Selecting an Oracle Communicator Account***



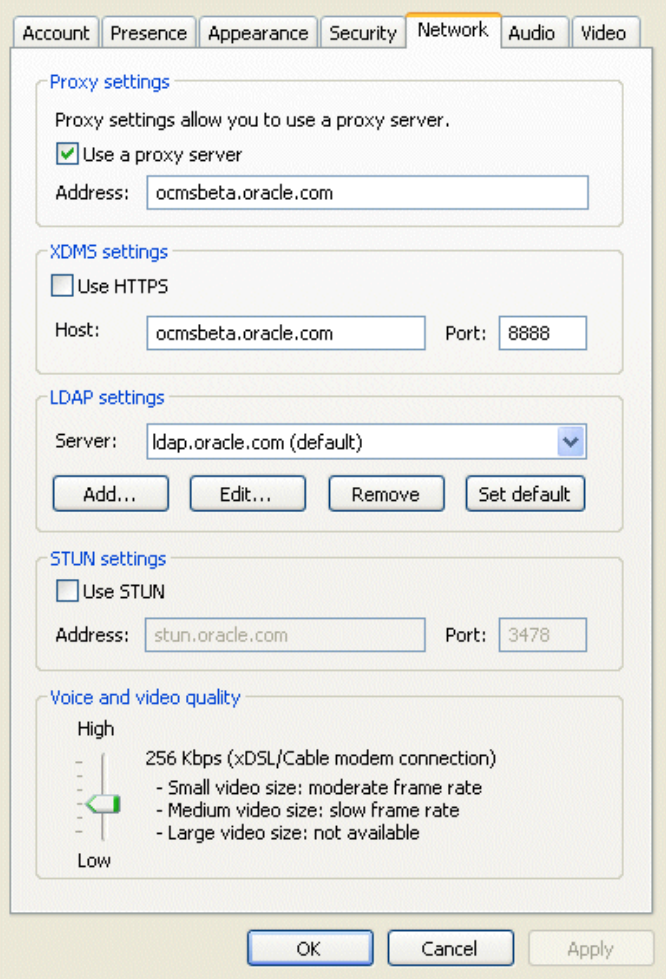

8. From the drop-down list, select the account you have just defined and click **OK**.
9. Enter your user name and password.
10. Click the settings icon and select **Preferences** > **Network**.
11. Check **Use a proxy server** and fill in the address of your SIP proxy server in the space provided (Figure 3–2) and click **OK**.
12. Check "Use HTTPS" if Oracle Application Server is configured with an HTTPS server.
13. Modify the default value under the section "XDMS Settings" from xcap.<machine>.<domain> to a valid value.

    For port, use one of the following values:

    - Use 8888 for standalone developer mode installation.
    - Use the HTTP server port for an Oracle AS installation with an HTTP server.
    - Use 7785 for an Oracle AS installation without an HTTP server.
    - For HTTPS, use the HTTPS port for the Oracle AS installation.

    This configuration allows Oracle Communicator clients to see each other's Presence information and to make calls.

***Figure 3–2 Configuring a Proxy Server***




14. Sign out of the Oracle Communicator client by right-clicking the client icon in the task bar and selecting **Sign Out**.
15. Log into Oracle Communicator with both clients and verify that you can send and receive messages between the clients.

    When you see a message in a client "Connected to example.com" accompanied by a green status LED, you are successfully connected.

## Install the Oracle Communicator FileTransferServlet

The FileTransferServlet enables file transfers between Oracle Communicator users. Although this servlet's EAR (Enterprise Archive) file ships on the same disc as OCMS, it is not included as part of the OCMS installation process because the FileTransferServlet does not generally reside on the same OC4J container as OCMS. As a result, the FileTransferServlet requires a separate installation, one accomplished by deploying its EAR file to an OC4J container using the wizard provided in the Application Server Control Console or the `admin_client.jar` command-line tool.

> **Tip:** In a high volume deployment, Oracle recommends that you deploy the FileTransferServlet to a different node to guarantee performance.

You can deploy the FileTransferServlet to any OC4J node that runs on Windows or Linux operating systems.

> **Note:** The OC4J node that hosts the `FileTransferServlet` must support both HTTP `GET` and `POST` operations.

For information on configuring Oracle Communicator to support file transfers, refer to *Oracle Communication and Mobility Server Administrator's Guide*. For more information on deploying applications to OC4J, refer to *Oracle Containers for J2EE Deployment Guide*.

## Verify Servlet Registration

The first interaction between a SIP client and the SIP servlet container is registration. Upon its first attempt to conceit to the SIP servlet container, the SIP client sends a `REGISTER` message to the server. If the SIP servlet container is correctly configured, it responds with `200 OK`, a standard message indicating that all is well.

If the SIP servlet container responds with `401 Unauthorized` this results in Oracle Communicator displaying a pop-up box for entering username and password. Upon entering the credentials, another REGISTER message is sent to the server, this time with authentication information.

You can watch this exchange by setting the log level to `info` (see "Set the Log Level") and watching the log as you run the SIP client.

To verify servlet registration:

1. Open a log window (make sure you have set the log level).
2. Select **Start > Programs > Oracle > Oracle Communicator**.
3. At the prompt, enter a user name and password (to provision a user, refer to "Provisioning Users and Applications" in the *Oracle Communication and Mobility Server Administrator's Guide*).
4. Run a separate instance of the SIP client (preferably on another computer) and login with a different user name and password.

   A sample log showing the `REGISTER` method and the SIP servlet container's response is as follows:

```
DEBUG [traffic] (NetworkEventWorker-11@) < REGISTER sip:example.com SIP/2.0
Call-ID: 77c49573-572e-46cc-893c-dc3b693eec90 CSeq: 1 REGISTER To:
<sip:alice@example.com> From: "alice@example.com"
<sip:alice@example.com>;tag=16feb568-cb96-4349-a825-a401e8189753 Max-Forwards:
```

```
70 User-Agent: OCMS-CallTron/4.5.7.1445 Contact:
<sip:alice@10.0.0.10:5062;transport=TCP>;q=1.00;agentid="4960c58b-286c-4426-bc3
7-b5c6d91dcd7d";methods="INVITE,NOTIFY,MESSAGE,ACK,BYE,CANCEL";expires=600
Content-Length: 0 Via: SIP/2.0/TCP
10.0.0.10:5062;branch=z9hG4bK-1b9f16b3-10f9-44eb-8d50-dedd2cda5ed3.1;rport

DEBUG [traffic] (NetworkEventWorker-13@) > SIP/2.0 200 OK Via: SIP/2.0/TCP
10.0.0.10:1267;received=10.0.0.10;branch=z9hG4bK-eb1c7aba-7aa6-44fc-98e8-e36567
13e09c.1;rport=1267 To: <sip:alice@example.com>;tag=6ed1a858-1083331acac--7ffa
From: "alice@example.com"
<sip:alice@example.com>;tag=89f9161e-762e-4750-9cea-970825f67848 Call-ID:
77c49573-572e-46cc-893c-dc3b693eec90 CSeq: 2 REGISTER Server:
OCMS-transactron/3.2.0-48 Content-Length: 0
```

## Test the Presence Server

Oracle Communicator enables testing the main functionalities of the Presence server. By running two instances of the SIP client with a different user logged into each, you can:

- Subscribe to a user's Presence by requesting permission to view the user's online status ("Subscribing to a User's Presence")
- Publish a user's Presence ("Test Publication of a User's Presence")
- Receive notifications of events, for example when a user logs on ("Test Receipt of Event Notifications")

### Subscribing to a User's Presence

A Presence subscription is a document that defines which users have access to a given user's Presence data, such as user status ("Available," "Away," and so on) or a "Gone fishing" message. A user must request permission to subscribe to a given user's Presence data. Once permission is granted, the subscription is saved and stored as a document.

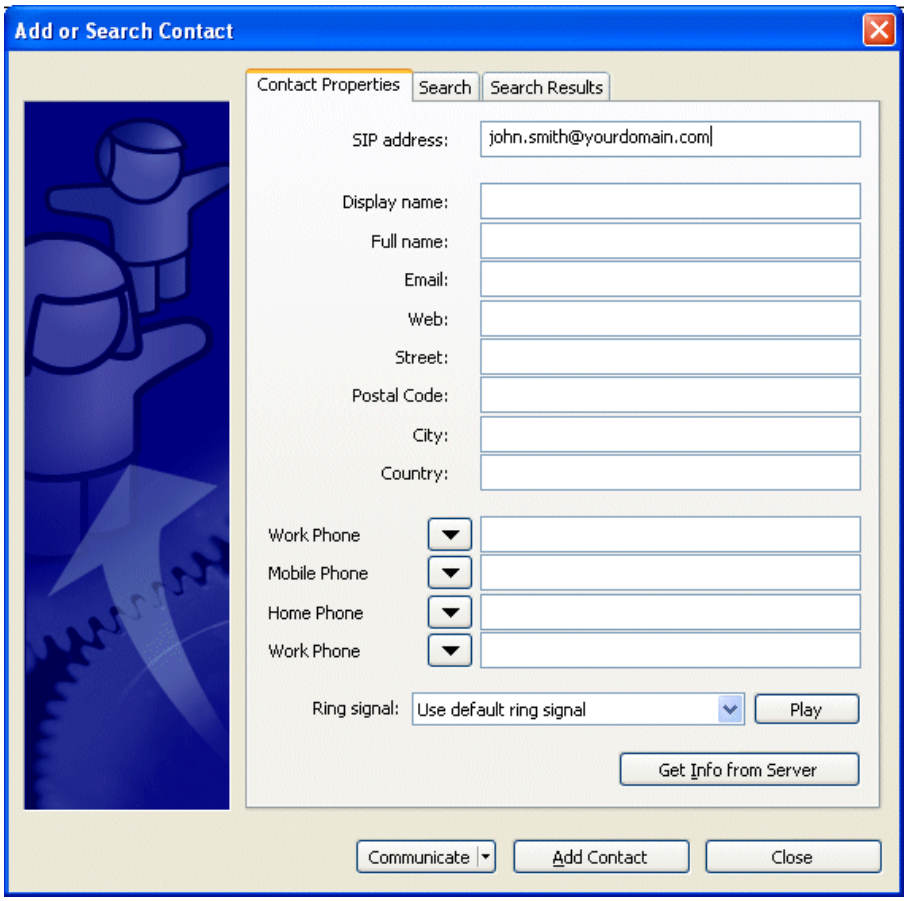
In this section, you use two instances of Oracle Communicator client to request and grant subscription to a user's Presence.

To subscribe to a user's Presence:

1. Make sure that each test user is logged into a separate instance of the Oracle Communicator client, preferably on different computers.
2. In the Oracle Communicator client where the first test user is logged in, click the **Contacts** button and select **Add Contact** (Figure 3–3).

> **Tip:** For more information on how to add a contact in Oracle Communicator refer to the Oracle Communicator online help.

*Figure 3–3 Adding an Oracle Communicator Contact*




3. In the Contact Properties tab that displays, enter the SIP address of the second test user you defined earlier in the space provided.
4. Click **Add Contact**.
5. Observe the second Oracle Communicator client instance. A message is displayed in which test user one requests permission to subscribe to the Presence of test user two.
6. In test user two's Oracle Communicator client, click **OK** to grant Presence subscription permissions to test user one and add the user to the contact list of test user two.

### Test Publication of a User's Presence

Oracle Communicator enables users to change their Presence from "Available" to "Away," for example. Once the selected Presence status is saved to the server, it is effectively published, enabling users subscribed to the publisher's Presence to view his Presence data.

To publish a user's Presence:

1. In the first test user's Oracle Communicator client, click the Presence Status drop-down list at the bottom of the client.
2. From the drop-down list, select **Away**.
3. Observe the second test user's Oracle Communicator client. The status of test user one should display as Away.

### Test Receipt of Event Notifications

You can test whether OCMS sends event notifications by logging off one of the test users and then logging the test user back in. As the second test user comes back online, a message displays on the first test user's computer, notifying the user that test user two is online.

To receive notification of an event:

1. Add test user two to test user one's contact list (see "Subscribing to a User's Presence" for details on how to add a user to the contacts list).
2. Log out test user two by clicking the settings icon and selecting **Sign out**.
3. Wait a moment, and then log test user two back in by selecting the user's account from the drop-down list and clicking **OK**.
4. Observe the computer running the Oracle Communicator client with test user one logged in.

   Oracle Communicator displays a message indicating that test user two has come online.

## Make a SIP Test Call

Test the functionality of OCMS by establishing a call from one test user to another. Run each instance of Oracle Communicator on a separate computer equipped with speakers and a microphone, or a multimedia headset.

To place a call:

1. In the first Oracle Communicator client, right-click test user two and click **Call**.
2. Answer the call from the second Oracle Communicator client by clicking **Answer Call**.
3. Begin talking to verify that the other party can hear you.

## Make a SIP to PSTN Test Call

If a SIP to PSTN gateway has been installed to route calls from the SIP network to the public switched telephone network (PSTN) you can make a test call from Oracle Communicator to a telephone number in the PSTN.

1. Launch an Oracle Communicator client.
2. In the Quickcall field, enter the calling information as follows:

   ```
   sip:<phone number>@<IP address of SIP to PSTN gateway>
   ```

   **Note:** You can substitute a hostname for an IP address if a DNS server is configured.

## Monitor OCMS Network Traffic with Ethereal

Ethereal® is a network protocol analyzer that you can use to monitor OCMS network traffic. Specifically you can monitor the TCP and UDP traffic (SIP clients use TCP or UDP to transport SIP messages) on the ports used for SIP, Presence, and Edge Proxy traffic.

Perform the following procedure to monitor OCMS network traffic with Ethereal:

1. Download and install Ethereal according to the instructions at the Ethereal website, `http://www.ethereal.com/`.

   When installing Ethereal you must also install the software package WinPcap for capturing live network data.

   Ethereal requires approximately 55 MB of hard disk storage to install.

2. Select **Start**, **Program Files**, and run **Ethereal**.
3. Configure Ethereal to monitor TCP and UDP traffic on the ports being used by OCMS. Typically you want to monitor ports 5060 through 5080. Refer to "Capture Filters" in the Ethereal help documentation.
4. Configure the capture output file.
5. Start capturing network traffic by selecting **Start** from the **Capture** menu.
6. View the captured data.

When troubleshooting OCMS network traffic issues, analyzing captured data by Ethereal is the primary method of analysis. A second source of data are the application server logs.

# 4

# Install Oracle Communication and Mobility Server with a Backend Oracle RAC Database

This chapter describes how to install and configure Oracle Communication and Mobility Server 10.1.3.4 to work with a backend Oracle RAC database. This document assumes that an *n*-node RAC database is installed, configured, and available for Oracle Communication and Mobility Server. This document does not describe how to install and configure the Oracle RAC database. Topics include:

- Gather RAC Information
- Install Oracle Communication and Mobility Server with RAC
- Post-Install Configuration of Oracle Communication and Mobility Server to Use the RAC Database

The instructions in this document apply to installing Oracle Communication and Mobility Server 10.1.3.4 to an existing instance of the Oracle Application Server. Oracle RAC database configuration is not supported for standalone (developer) installations of Oracle Communication and Mobility Server. Also, connection caching, fast connection failover, and such advanced configuration features for the Oracle RAC database are not covered in this document.

## Gather RAC Information

You must have the following information on hand regarding the Oracle RAC database:

- Database name
- Number of nodes in the RAC database
- For each node you need the following information:
  - Host name
  - Port number
  - Oracle SID
  - ORACLE_HOME and ORACLE_CRS_HOME
  - Password for sys user

In addition to the hostnames, production deployments typically also have virtual IP addresses for each DB node. Those hostnames corresponding to the virtual IP addresses should be used during install and configuration instead of the hostnames.

# Install Oracle Communication and Mobility Server with RAC

Follow these steps to install Oracle Communication and Mobility Server and configure it to use a backend RAC:

- Create Services on RAC Database
- Install Oracle Communication and Mobility Server

## Create Services on RAC Database

Before installing Oracle Communication and Mobility Server, you must create a database service on the RAC database which will be used by Oracle Communication and Mobility Server to connect to the database. Create this database service so that it is available on all nodes in the Oracle RAC environment.

On any of the RAC nodes, run the following commands:

```
cd $ORA_CRS_HOME/bin
./srvctl add service -d <database name>
                     -s <service name>
                     -r <comma separated list of
                         ORACLE_SIDs for the RAC nodes>
```

To check the status of the service you just created, run:

```
./srvctl status database -d <database name> -s <service name>
```

You will receive this response:

```
Service <service name> is not running.
```

You will find that the new service is not running; you must explicitly start the new service by issuing the following command:

```
./srvctl start service -d <database name> -s <service name>
```

The following response will be returned:

```
Service <service name> is running on instance(s) (inst. 1, inst. 2, etc.)
```

You are now ready to start the installation of Oracle Communication and Mobility Server.

## Install Oracle Communication and Mobility Server

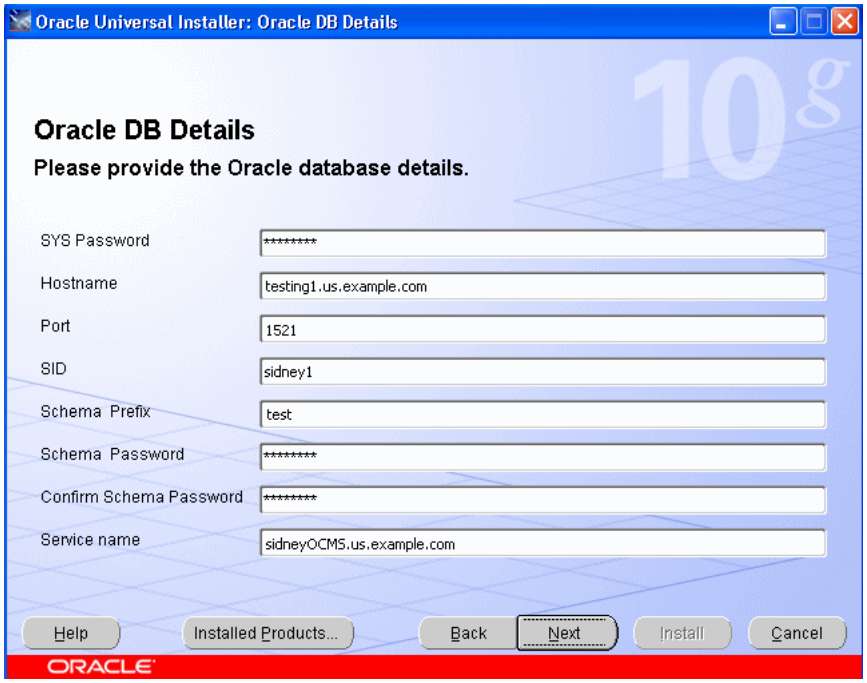
Oracle Universal Installer does not support configuring Oracle Communication and Mobility Server with a RAC database at install time. However, you can configure Oracle Communication and Mobility Server with a RAC database by using the following workaround:

- Install Oracle Communication and Mobility Server by pointing to one of the nodes of the RAC database at install time.
- Modify the data sources post-install to configure Oracle Communication and Mobility Server with the n-node RAC instance.

Start the installation of Oracle Communication and Mobility Server to an existing installation of Oracle Application Server (see Install Oracle Communication and Mobility Server for details) and proceed until you reach the database configuration screen. On this screen, enter information corresponding to any one database node that is part of the RAC database service you created.

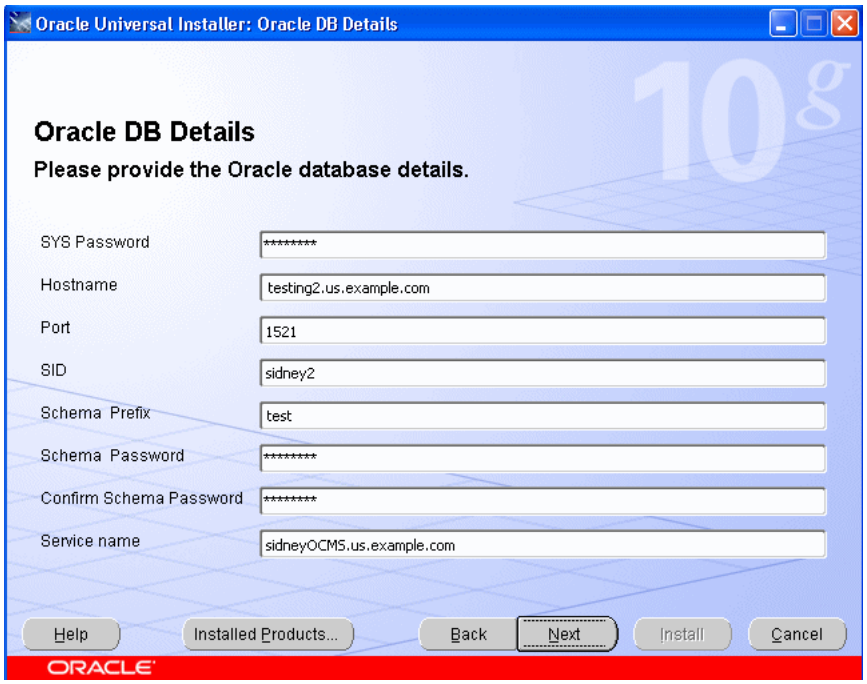
For instance, the following DB configuration screen (Figure 4–1) shows information corresponding to the first DB node.

***Figure 4–1 Database information for first DB node***




Alternatively, you can provide information corresponding to the second (or other) node in your database service as shown in Figure 4–2:

***Figure 4–2 Database information for a subsequent DB node***



When you click **Next**, you will see a screen prompting you to specify the location of the SDP datafiles. Choose **No** on this screen and let the SDP datafiles be created in the default datafile location (on the shared network drive) of the backend RAC database.

Click **Next** and complete the Oracle Communication and Mobility Server installation.

Your Oracle Communication and Mobility Server schemas are now accessible from any of the nodes in your database service. You can verify this by using SQLPLUS to connect to any of the nodes in the database service (in this case on the machine named *testing1.us.example.com*:

```
cd $ORACLE_HOME/bin
./sqlplus /nolog
SQL> conn userx_
orasdpsds/myPassword1@testing1.us.example.com:1521/sidneyOCMS.us.example.com
SQL> select * from tab;
```

Verify that you can see your tables and exit out of SQLPLUS. Now connect to any other node from this machine, for instance, *testing2.us.example.com*:

```
./sqlplus /nolog
SQL> conn userx_
orasdpsds/myPassword1@testing2.us.example.com:1521/sidneyOCMS.us.example.com
SQL> select * from tab;
```

Verify that you can see your tables over this connection as well.

If your Oracle Communication and Mobility Server installation was successful, you have a working Oracle Communication and Mobility Server environment. However, the database connections from Oracle Communication and Mobility Server are to only one configured database node of the RAC database. A post-install configuration step is required to modify the database connections to the RAC database.

## Post-Install Configuration of Oracle Communication and Mobility Server to Use the RAC Database

On your Oracle Communication and Mobility Server server machine, do the following:

```
cd $ORACLE_HOME/j2ee/ocms/config
```

Edit `data-sources.xml` at this location.

Change the JDBC URL for the *subscriber data services*, *location service*, and *XDMS* from:

```
url="jdbc:oracle:thin:@//testing1.us.example.com:1521/sidneyOCMS.us.example.com"
```

to

```
url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_
BALANCE=on)(ADDRESS=(PROTOCOL=tcp)(HOST=testing1.us.example.com)(PORT=1521))(ADDRE
SS=(PROTOCOL=tcp)(HOST=testing2.us.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=tcp)
(HOST=testing3.us.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=tcp)(HOST=testing4.us
.example.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_
NAME=sidneyOCMS.us.example.com)))"
```

Replace the *hostnames*, *port numbers*, *database service name*, and *number of instances* with values corresponding to your RAC database.

Save `data-sources.xml` and restart your server.

To verify that your Oracle Communication and Mobility Server database connections are going to all nodes of your RAC database, issue the following command:

```
netstat -a | grep 1521
```

and verify that TCP connections are established from your Oracle Communication and Mobility Server server to all the RAC nodes (on port 1521 in this example).

Instead of editing the `data-sources.xml` file, you can make changes to the Oracle Communication and Mobility Server JDBC resources using Oracle Enterprise Manager.

# 5

# Presence Large Deployment Installation

This chapter describes how to complete a Presence Large Deployment. Topics include:

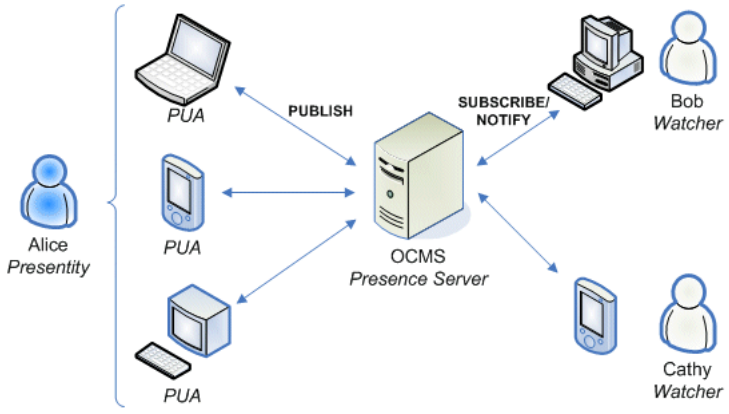


## Introduction

OCMS includes its own presence server allowing clients to publish their presence information such as *busy*, *available*, *in a meeting* and so on, and have that information displayed to those who have registered an interest in knowing when your presence state changes. From an end-client perspective, the Multi Node Presence Server is no different from Single Node; to end users, it is a black-box and they cannot notice a difference.

This section describes the basic Presence functionality, and how to scale the Presence Server by using a new element in OCMS 10.1.3.4, namely the *User Dispatcher*. To learn more about how the Presence Server works in general, see *Oracle Communication and Mobility Server Administrator's Guide*.

***Figure 5–1 General usage of the Presence Server***




## Definitions

There are quite few new concepts that need to be understood in order to successfully install and manage a large scale presence service. This section defines and explains the major concepts.

***Table 5–1 Major Presence terms***

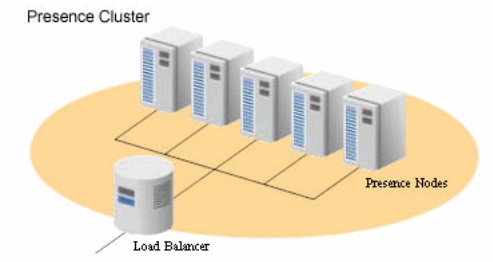
| Term | Definition |
|---|---|
| Presence Cluster | A cluster consisting of X number of Presence Nodes front-faced by Y number of Load Balancers. See Presence Cluster for more information. |
| Presence Node | A physical machine (a node) consisting of one User Dispatcher front-facing X number of Presence Server instances (PS). |
| Presence Server (PS) | The actual Presence Server (instance) responsible for processing subscribe-and-publish requests made to the presence event-package as well as subscribes made to `presenence.winfo`. The Presence Server is not to be confused with the Presence Service. |
| Presence Service | The term *Presence Service* is defined in RFC 2778 and defines at the very highest-level a service that processes all related presence traffic. This includes traffic for watcher info, traffic for determining a watchers permission to subscribe to a Presentity and the actual presence updates and so on. |
| XDM | XML Document Management |
| XDMC | XDM Client – any client that access a XDM network. Since the Presence Server instances access the XDM Cluster, they are also acting as XDM Clients. |
| XDM Cluster | A cluster consisting of X number of XDM Nodes, front-faced by Y number of Load Balancers. See XDM Cluster for more information. |
| XDM Node | A physical machine (a node) consisting of one Aggregation Proxy, one User Dispatcher and X number of XDM Servers (XDMS). |
| XDM Server (XDMS) | The actual XDMS instance responsible for storing XML documents and allows for query and manipulating of those documents through XCAP. The XDMS also allows for subscription to changes in those documents using SIP SUBSCRIBE/NOTIFY. The XDMS instance plays the same role in the XDM Node as the Presence Server instance does on the Presence Node. |

## Presence Cluster

The Presence Cluster is a set of Presence Nodes front-faced by one or more Load Balancers as illustrated in Figure 5–2. The Presence Cluster is responsible for processing incoming subscribe and publish requests made to the presence event-package, for sending out NOTIFY:s whenever appropriate. The Presence Cluster also accepts and processes subscribe requests for the `presence.winfo` event-package.

The Presence Cluster interacts with the XDM Cluster to obtain information needed to complete its responsibilities. The information queried from the XDM Cluster includes users' *presence rules* and *pidf-manipulation* documents (that is, the users' hardstates).

***Figure 5–2 Presence Cluster***

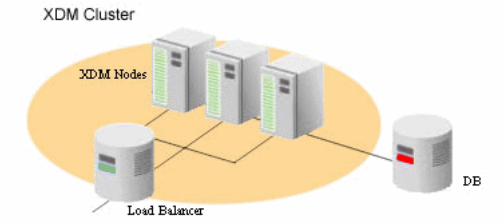


### XDM Cluster

The XDM cluster is a set of XDM Nodes front-faced by one or more Load Balancers as shown in Figure 5–3.The XDM cluster processes all XDM related traffic (that is, SIP subscribe traffic towards the *ua-profile event-package* and XCAP traffic). As such, it processes everything that has to do with manipulating XML documents. The XDM Cluster uses a database for storage of the XML documents but the database (and potentially its cluster), are NOT part of the XDM Cluster.

Since the XDM Cluster processes all XML documents, each node will be both a Shared XDMS, and a PS XDMS.

***Figure 5–3 XDM Cluster***




## Presence Multi-Node Topology

Figure 5–1 illustrates how clients interact with a single Presence Server. In order to handle a larger user base, this single server must be scaled out. This is accomplished by adding more Presence and XDM nodes to the system. This scaled-out Presence Service is divided into two distinct clusters: the *Presence* and *XDM* clusters as defined in Definitions.

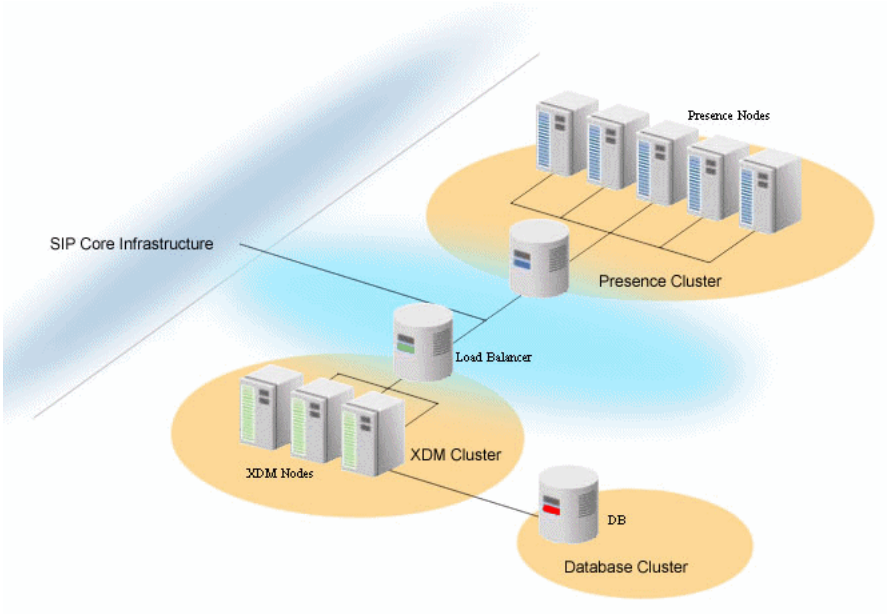
---

**Note:** It is important for administrators to understand that the final Presence Service consists of multiple distinct clusters, nodes, and components, but to end users, this fact is invisible.

---

Figure 5–4 shows a complete Presence and XDM cluster with all necessary components. This figure also illustrates that the two clusters, *Presence* and *XDM*, are treated as two separate clusters, and the way into those two networks for initial traffic is always through their respective *Load Balancers*. Even the Presence Servers will actually go through the load balancers of the XDM Cluster when setting up subscriptions towards (for example) a presence rules document. However, once a subscription has been established, subsequent requests will not go through the load balancer, but rather directly to the XDMS instance hosting the subscription. All nodes in the XDM Cluster are directly accessible from the Presence Cluster. A PS will actually go directly to an XDMS instance when fetching a presence rules document.

*Figure 5–4 Two clusters in a large deployment*




Note that even though this image shows two different Load Balancers, one in front of the Presence Cluster and one in front of the XDM Cluster, they typically are the same physical box.

## Components Overview

Each of the two different nodes, the Presence and XDM Node, consists of a set of smaller components. These components are defined and discussed in this section, and it is important to understand the difference and purpose of these components. When performing the actual installation, these various components are the artefacts that will be deployed onto the physical nodes.

### Load Balancer

The purpose of the *Load Balancer* is to distribute traffic across the other components. Looking at the low-level components, a load balancer will always distribute SIP traffic to a User Dispatcher, but for XCAP traffic it will distribute the traffic to an Aggregation Proxy.

### User Dispatcher

The job of the *User Dispatcher* is to extract the user identity of the incoming request and based on that user, dispatch the traffic (both SIP and XCAP) to either a PS or an XDMS depending on the sub-application.

### Presence Server

The *Presence Server* is the component responsible for processing incoming SUBSCRIBE and PUBLISH requests to the presence event-package and to send out a NOTIFY whenever appropriate. It also processes incoming SUBSCRIBE requests to the `presence.winfo` event-package. The PS interacts with the XDMS in order to get hold of presence rules and `pidf-manipulation` (presence hardstate) documents.

### XDM Server

The main purpose of the XDMS is to act as a remote file storage of XML documents. Those documents can be manipulated using XCAP which also exposes a SIP interface for allowing clients to set up subscriptions for changes in a document. The event package is *ua-profile*, and the XDMS will convey the state of the document by sending out NOTIFY:s to the subscribers.

### Aggregation Proxy

The role of the *Aggregation Proxy* is to authenticate all incoming XCAP traffic before it proxies those requests to the User Dispatcher. As such, the Aggregation Proxy will never directly access any XDMSs; the User Dispatcher will be responsible for doing that.

### Database

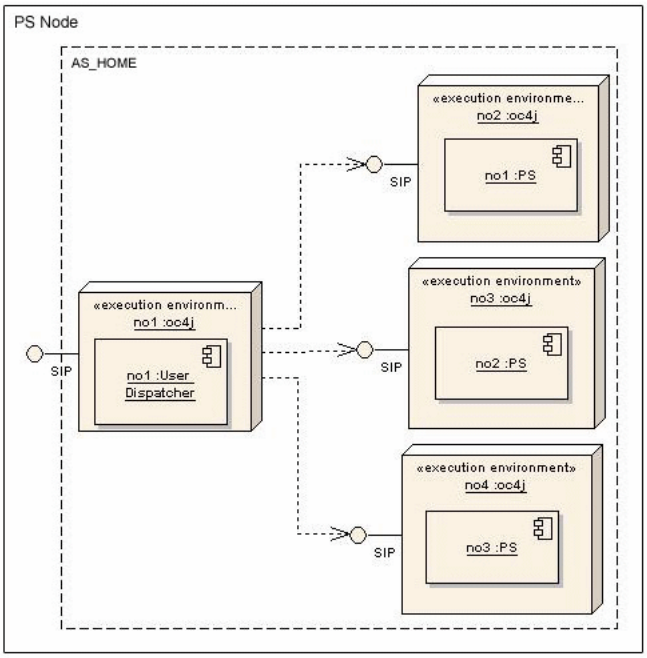
The database is where the XML documents managed by the XDMS are physically stored.

## The Presence Node

The *Presence Node* is the main component in the Presence Cluster and is responsible for dispatching incoming traffic to the correct Presence Server instance and, from a black-box perspective, servicing users with presence information. It is important to understand that the User Dispatcher serves the same purpose both in a single node deployment and in a multi-node deployment (that is, its purpose is to dispatch incoming traffic to a particular PS instance and if this instance is running on the same physical node or not is of no relevance to the User Dispatcher). The User Dispatcher identifies a particular node by its full address (IP-address and port), and has no concept of *local instances*.

Figure 5–5 shows the layout of a typical Presence Node. The node will always have a User Dispatcher deployed that serves as the main way into the node itself. Typically, the User Dispatchers would listen to port *5060* (the default port for SIP) and the other Presence Servers on that node would listen on other ports. In this way, a single node will appear as one Presence Server to clients but is in fact multiple instances running behind the User Dispatcher. Each of the components deployed on the Presence Node is executing in their own separate JVM (that is, the User Dispatcher and the PS instances are all executing in their own OC4J instances).

***Figure 5–5 Components deployed onto a Presence Node***




Note that all of these OC4J instances (four in the example above) are executing within the same Oracle Application Server (same AS_HOME).

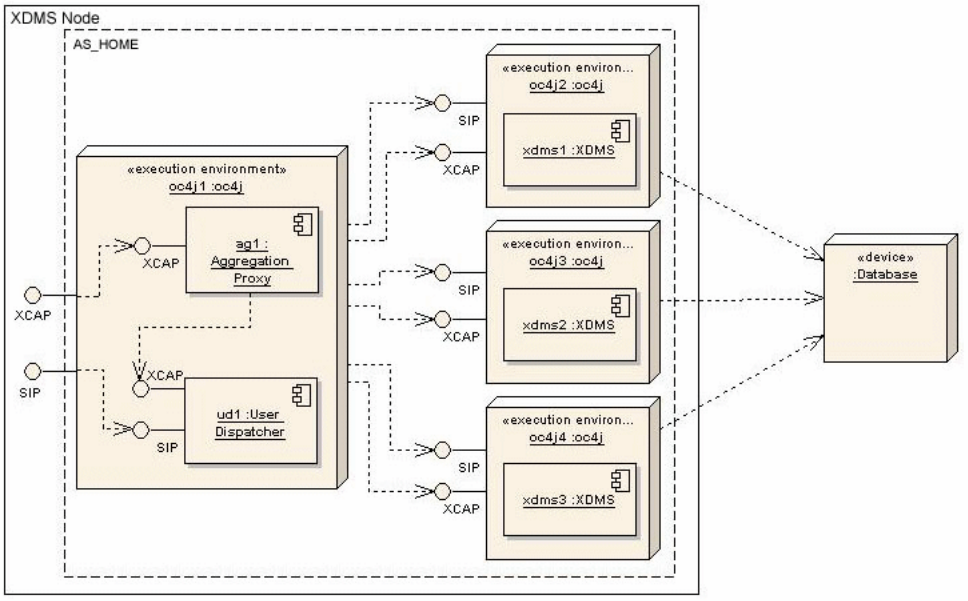
## The XDM Node

The XDM Node, as Figure 5–6, will always have an Aggregation Proxy deployed that typically would be listening on port *80* for XCAP traffic (XCAP goes over HTTP). The Aggregation Proxy will authenticate incoming traffic and upon successful authentication forward the request to the User Dispatcher. As with the Presence Node, the XDM Node will also have a User Dispatcher deployed (usually on port 5060) and for SIP traffic there is absolutely no difference between the XDM and Presence Nodes. The difference between the two types of nodes is that the User Dispatcher will also participate in dispatching XCAP traffic. Hence, just as it does with SIP, it extracts the *user id* out of the request, and based on that, maps the request to a particular XDMS instance to which it forwards the request.

Further, there will be X number of XDMS instances deployed to which the User Dispatcher dispatches both SIP and XCAP traffic. Just as in the case of the PS instances on the Presence Node, each XDMS instance is not aware of the others and is executing in isolation.

Also note that the Aggregation Proxy and User Dispatcher are deployed onto the same OC4J container and will therefore also be using the same JVM, but all OC4J instances are still (just as in the case of the Presence Node), executing within the same Oracle Application Server.

*Figure 5–6 XDMS Node*



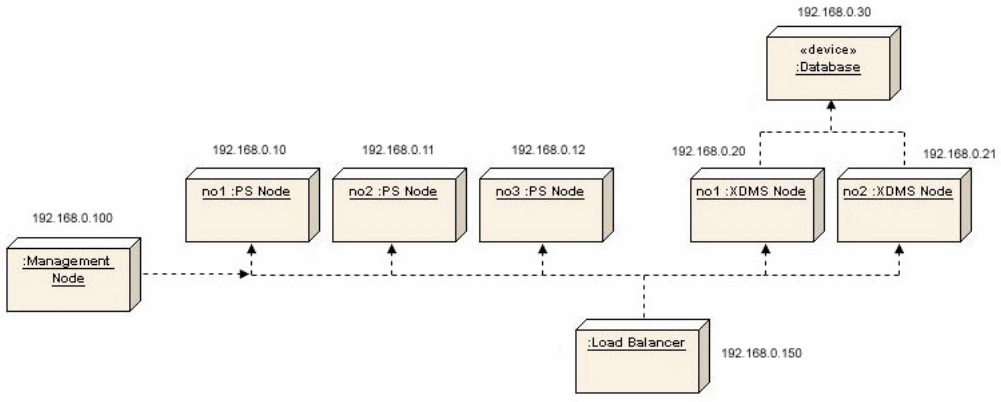

# Installation

The previous sections described the general layout and components of the Large Presence Deployment. This section describes how to install such a system.

## Example Network

In order to easier explain the necessary steps, the network shown in Figure 5–7 will be used as an example.

*Figure 5–7 Example network*




The network consists of three PS Nodes that together form the *Presence Cluster*. The XDM Cluster consists of two XDMS Nodes and are accessing a database with the address *192.168.0.30*. Both of the clusters are sharing the same physical Load Balancer. Note that the Load Balancer does not divide the network into any external and

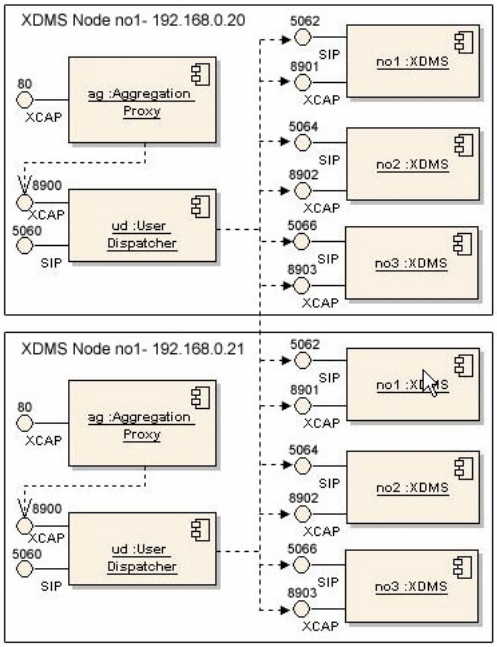
internal networks and as such it really only has one "leg" that is sitting on the IP address *192.168.0.150*.

For easy manageability, one Management Node has been added to the network and it is through this node that configuration of all other nodes and instances will be performed.

Going into the specifics of the two clusters, there is the Presence Cluster consisting of three Presence Nodes as previously mentioned. Figure 5–8 shows a more detailed view of the Presence Cluster. Each Presence Node has one User Dispatcher deployed and three PS instances. These four components are executing within their own OC4J instance but these details have been left in order to enhance readability.

Figure 5–8 shows the detailed view of the XDM Cluster and in this particular example there are three XDMS instances running front-faced by one User Dispatcher and one Aggregation Proxy. As pointed out in The XDM Node, the Aggregation Proxy and the User Dispatcher will be executing on the same OC4J instance whereas the three XDMS instances will be running on their own OC4J. All of these OC4J instances (four of them) are running within the same Oracle Application Server.

***Figure 5–8 Cluster example***




## Install Oracle Application Server 10.1.3.4

For every node that will be installed into the final system (Presence Node, XDM Node or a Management Node), they will all have the same basic installation (that is, they will all run the Oracle Application Server 10.1.3.4). As such, the following section details how to install and set up the Oracle Application Server. The first step is to install Oracle Application Server 10.1.3.2 and then apply the 10.1.3.4 patch set. These are the main steps:

1. Start Oracle Universal Installer to install Oracle Application Server 10.1.3.2.
2. Choose **Advanced Installation**.
3. Choose **Oracle WebCenter Framework** (second from the bottom) as the installation type.

Unless the node you are installing is the Management Node, do not check the check-box that reads: *Start Oracle Enterprise Manager 10g ASControl in this instance*. There will only be one node in the system where the ASControl will run and that is on the Management Node. All other nodes in the system (PS and XDMS Nodes) will be controlled through the Management Node.

4. Enter a discovery address and ensure that the value you use for the multicast address is the same for all nodes in the cluster. More specifically, the PS and XDMS Nodes must have the same discovery address as the Management Node, otherwise they will not be detected by it. For more information, see your Oracle Application Server installation documentation.

Once the installation for AS 10.1.3.2 is complete, continue with the following steps to install the AS 10.1.3.4 patch set:

1. Apply the 10.1.3.4 WebCenter patch by running Oracle Universal Installer for the 10.1.3.4 patch.
2. Install Java 5 update 14. The recommended way to do this is to follow these steps:
   - Run the Sun Java installer for JDK 1.5 update 14 to install the JDK to a directory of your choice that refers to `<jdk-directory>`.
   - Go to $ORACLE_HOME and back up the JDK installed there by renaming the file `jdk` to `jdk.install.backup`.
   - Create a symbolic link named `jdk` in $ORACLE_HOME that points to <jdk-directory>.

## Install the Management Node

Through the Oracle Application Server Control, it is possible to configure and maintain all the nodes in a cluster. As such, having one Management Node in the cluster will ease the operation of the system and allow for fast changes in the configuration that will take effect across all nodes immediately.

Installing this management node is done by installing Oracle Application Server on one node (in our network example this node is running on *192.168.0.100*), and enable it to run the Application Server Control. Ensure the check-box *Start Oracle Enterprise Manager 10g ASControl in this instance* is checked. This is all it takes to install the Management Node.

All nodes in the system must be configured to use the same discovery address and in this example we used the multicast address *235.0.0.1:6789*. Hence, this is the address that must be used for all our nodes in the system.

The Management Node should also contain the `setupLinux.tar.gz` file (unpacked). This will ensure that .ear files that are to be deployed later are already located on the management node, and are ready for deployment.

## Install the Presence Nodes

The following steps must be completed in order to install one Presence Node. Of course, there will be many Presence Nodes within a Presence Cluster, and each of the these steps must be repeated for each of those nodes.

**Note:** All nodes should be installed and configured at the same time to ensure smoother deployment.

The steps to take are as follows:

1. Install Oracle Application Server 10.1.3.4.
2. Using Oracle Universal Installer, install a first instance containing the User Dispatcher. The SipContainer is installed by default.
3. Create as many extra OC4J instances as needed.
4. Configure these OC4J instances.
5. Deploy and configure presence onto those newly created OC4J instances.
6. Configure the User Dispatcher.
7. Post-Installation/Tuning the Installation.

### Install Oracle Application Server

Each Presence Node is executing on top of the Oracle Application Server and as such must be installed first. The necessary steps are outlined in Install Oracle Application Server 10.1.3.4 but ensure that you do not check the *Start Oracle Enterprise Manager 10g AsControl in this instance* box since this is not a Management Node.

For the discovery address you must ensure that you choose the same address as you chose for the Management Node. In our example network, the discovery address chosen for the Management Node was *235.0.0.1:6789* so this is what will be used in this example.

### Install User Dispatcher

Follow these steps to install User Dispatcher:

1. Using Oracle Universal Installer for OCMS, select only *User Dispatcher* and the *SipContainer*.
2. Use the default SIP port value of *5060* for this instance.
3. Complete the installation steps in Oracle Universal Installer.

The Installer creates an OC4J instance named *ocms* executing in your Oracle Application Server. The only application deployed on this instance is User Dispatcher.

### Create More Instances

The number of instances you create depends on the available memory. Each new instance will be set up to consume 2.5 GB of memory and as such, the broad rule of thumb is to create as many instances that fit within the memory available. Note that there must still be some memory left for the operating system.

Final tuning can be performed afterwards to optimize system performance.

Before continuing, shut down running instances: `$ORACLE_HOME/opmn/bin/opmnctl stopall`

Use the `createinstance` command in `$ORACLE_HOME/bin` to create more OC4J instances. For instance:

```
cd $ORACLE_HOME/bin
./createinstance -instanceName ps1 -groupName presence -httpPort 8901
-defaultAdminPass
```

where *ps1* is the name of the instance, and the group is named *presence*. If the group did not exist, then it will be created.

*-httpPort 8901* specifies the port number, and Oracle recommends using consecutive available ports for ease of management. In our example network, we use *8901*, 8902 and 8903 for the OC4J instance ps1, ps2 and ps3 respectively.

*-defaultAdminPass* omits setting a password at this time. This directs `createinstance` to omit setting a password for the instance. Later you will issue another command to set the password for the instance. If you do not include this option, `createinstance` will prompt you for a password. If you are creating just a few instances, you can do it this way, but if you want to execute a script for instance creation, it is more efficient to use *-defaultAdminPass*.

To set the password for the instance you just created, execute this command:

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../<instance-name> -jar jazn.jar -activateadmin
<password>.
```

For example, to set the password for the *ps1* instance to *myPassword1*, you would execute:

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps1 -jar jazn.jar -activateadmin myPassword1
```

Repeat these two commands as many times as needed in order to create enough instances. In our example network, we need to run the pair of commands three time, to create three OC4J instances ps1, ps2 and ps3 for the three presence instances respectively:

```
cd $ORACLE_HOME/bin
./createinstance -instanceName ps1 -groupName presence -httpPort 8901
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps1 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName ps2 -groupName presence -httpPort 8902
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps2 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName ps3 -groupName presence -httpPort 8903
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../ps3 -jar jazn.jar -activateadmin myPassword1
```

### Configure OC4J Instances

The previous step outlined how to create new OC4J instances to which the actual Presence Server will be deployed. Before you can deploy, you must configure those new instances to also pick up the Sip Servlet Container. The necessary jars have already been installed into the Oracle Application Server when you installed the User Dispatcher, so all you need to do is to further configure each of those *ps X* instances to pick up the Sip Servlet Container jars from the shared library. You must also configure the new OC4J instances for proper startup and logging. These are the overall steps you must perform for all the new OC4J instances:

1. Configure the instances to pick up the shared library where all the necessary jars for the Sip Servlet Container reside. This involves editing the `boot.xml` file.
2. Configure logging. For your new instances to use the logging done by the XDMS instance, you must edit the `j2ee-logging.xml` file.

3. Edit the start-up and shut-down parameters for the instances so that the Sip Servlet Container is loaded. This is done by editing the `$ORACLE_HOME/opmn.xml` file.
4. Specify the SIP ports to which the new instances should be listening.
5. Configure the Sip Servlet Container to listen to the correct IP address.
6. Add the `xcap config` directory.
7. Verify your configuration.

**Configure the instance to use the shared libraries** Configure the instances to pick up the shared library where necessary jars for the Sip Servlet Container reside. To achieve this, copy the `boot.xml` from the *ocms* instance that was created by the OCMS installer:

```
cp $ORACLE_HOME/j2ee/ocms/config/boot.xml $ORACLE_HOME/j2ee/<instance
name>/config/
```

This copies the correctly-configured boot.xml found in the ocms instance into another instance. Repeat this for all the newly created instances. In our example network, we issue the above command three times, replacing *<instance name>* with *ps1*, *ps2* and *ps3* respectively to copy boot.xml into the three OC4J instances.

**Configure logging** Just as you copied the boot.xml file to load the shared libraries you can copy the configuration file for logging. Copy the `j2ee-logging.xml` file found in the config directory of ocms over to all your instances:

```
cp $ORACLE_HOME/j2ee/ocms/config/j2ee-logging.xml $ORACLE_HOME/j2ee/<instance
name>/config/
```

In our example network, we issue the above command three times, replacing <instance name> with *ps1*, *ps2* and *ps3* respectively to copy `j2ee-logging.xml` into the three OC4J instances.

**Configure JVM start/stop parameters** Configure the JVM parameters to set the start-up and shut-down of the instances so that the Sip Servlet Container is loaded. This is done by editing the `opmn.xml` file to set the start and stop parameters on all the presence OC4J instances (*ps1*, *ps2* and *ps3* in our example network) as well as the ocms instance.

Start parameters:

```
<data id="java-options" value="-server -Xmx2500M -Xms2500M -Xloggc:/<instance
name>/sdp/logs/gc.ps1.log -XX:+PrintGCDetails -XX:NewRatio=3
-XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -Dhttp.maxFileInfoCacheEntries=-1 -Djava.security.policy=$ORACLE_
HOME/j2ee/<instance name>/config/java2.policy -Djava.awt.headless=true
-Dhttp.webdir.enable=false -DopmnPingInterval=1
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.s
ipservletcontainer.deployer.Oc4jApplicationHook "/>
```

Stop parameters:

```
<data id="java-options" value="-Djava.security.policy=$ORACLE_HOME/j2ee/<instance
name>/config/java2.policy -Djava.awt.headless=true -Dhttp.webdir.enable=false
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.s
ipservletcontainer.deployer.Oc4jApplicationHook"/>
```

The meaning of these parameters is explained below:

*-Xmx2500M* Set the maximum JVM memory to 2.5GB

*-Xms2500M* Set the minimum JVM memory to 2.5GB

*-XX:+PrintGCDetails* Enable logging of collection activity

*-XX:NewRatio=3* Set the ratio between the young generation and the old generation of the heap to 1:3 (in other words, the combined sizes of the eden space plus survivor space is one fourth of the total heap size).

*-XX:+UseConcMarkSweepGC* Enable the concurrent mark-and-sweep garbage collector (also known as the concurrent low pause collector).

*-XX:+UseParNewGC* use parallel threads.

*-XX:PermSize=128m* Set the initial size of the permanent generation to 128MB.

*-XX:MaxPermSize=128m* Set the maximum size of the permanent generation to 128MB.

*-Xss128k* Set the stat size for each thread to 128 KB.

*-Dhttp.maxFileCacheEntries=-1* Disable caching on the HTTP server that is bundled with the Oracle Application Server. In this release, caching must be disabled in order to achieve good performance for the Presence Server.

*-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipservletcontainer.deployer.Oc4jApplicationHook* Enables OC4J to load the SipContainer.

> **Note:** The ocms instance you installed does not contain all parameters needed for a multiple JVM installation, so make sure that the above parameters are set on the ocms instance as well.

**Configure the SIP Ports** Now that you have set the start and stop parameters, you must configure the Sip Servlet Container to listen on the correct ports. By default, the Sip Servlet Container will listen to port *5060* for SIP and *5061* for SIPS. If you look at the file `$ORACLE_HOME/opmn/conf/opmn.xml`, you will see that ocms instance that was created by the OCMS installer is configured with these default values as shown below:

```
<port id="sip" range="5060"/>
<port id="sips" range="5061"/>
```

Those values for the ocms instance should remain as-is, since you want the User Dispatcher to listen for incoming traffic on those ports. For the other OC4J instances that you created for presence nodes, you must configure different ports for each of them to avoid port conflicts. For ease of management, it is recommended to use available ports in series. In our example network, we assign the ports for the presence instances beginning with *5062* – yielding the following configuration:

*ps1*

```
<port id="sip" range="5062"/>
<port id="sips" range="5063"/>
```

*ps2*

```
<port id="sip" range="5064"/>
<port id="sips" range="5065"/>
```

*ps3*

```
<port id="sip" range="5066"/>
<port id="sips" range="5067"/>
```

**Configure the Sip Servlet Container to listen ton the correct IP address** By default the Sip Servlet Container will listen on IP address *127.0.0.1*. You must change that to be the externally-addressable IP address of the machine. In the previous steps you configured the Oracle Application Server to load up the container but do not have any configuration files for the actual Sip Servlet Container itself. In order to generate the default configuration files, you must start the Oracle Application Server and then stop it again. Once the default configuration files are generated, you can edit them to suit your deployment.

To start the Oracle Application Server:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

Once it is up and running, shut it down:

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

A new configuration directory named *sdp* now exists under `$ORACLE_HOME/j2ee/<instance name>/config/sdp/`. Here you will find all the information related to configuration of OCMS on the respective instances. Each instance has a file named `SipServletContainer.xml`. You must edit this file for each of the instances and change the IPAddress attribute from 127.0.0.1 to the IP Address at which your Sip Servlet Container should listen. However, since this file should be identical for all Sip Servlet Containers on this machine, you do not have to edit each one manually – instead, copy the one found under the *ocms* directory and reuse it. Here is the command:

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/SipServletContainer.xml $ORACLE_
HOME/j2ee/<instance name>/config/sdp/
```

Replace <*instance-name*> with the name of the presence OC4J instances. In our example network, we execute the above command three times, once each for *ps1*, *ps2* and *ps3*. Now that the Sip Servlet Container is configured for all the instances, you can start the server. To start all the OC4J instances on the Oracle Application Server:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

**Configure the xcap config directory** Even though the `ua-profile` event package is not explicitly used on the presence nodes, it must be configured to be present; therefore we need to you must copy the xcap configuration files to all the presence OC4J instances. To do so, copy the xcap directory into the sdp config directory `$ORACLE_HOME/j2ee/<instance-name>/config/sdp/` for each instance. In our example network, we must copy the xcap directory into:

```
$ORACLE_HOME/j2ee/ps1/config/sdp/
$ORACLE_HOME/j2ee/ps2/config/sdp/
$ORACLE_HOME/j2ee/ps3/config/sdp/
```

You will find the xcap directory under the directory where you extracted the OCMS installer. If, for instance, you extracted the installer into the directory `OCMS_INSTALLER`, then the xcap directory will be under:

```
OCMS_
INSTALLER/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/DataFiles/Expanded/Shipho
me/shiphome-archive/shiphome-archive/xcapconf/conf
```

**Configure springbeans.xml** Copy the `springbeans.xml` from the sdp config directory of the ocms instance into the sdp config directory of all the presence instances:

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/springbeans.xml $ORACLE_HOME/j2ee/<instance
name>/config/sdp/
```

In our example network, we execute the above command three times, replacing <instance-name> with *ps1*, *ps3* and *ps3* respectively.

**Verify your configuration settings** Before continuing with the installation, verify that your actions were successful by examining the logs. You can also issue the `netstat` command to verify that the server is listening on the correct ports.

**Log Files:** You should also regularly monitor the log files; they will reveal if anything is wrong with the configuration. There are two main log files you should pay attention to. The first one is the console output from each of the instances, which is located at `$ORACLE_HOME/opmn/logs` and a typical file will be named like so for the presence nodes:

```
presence~<instance-name>~default_group~1.log
```

You will also find the one for the ocms instance as well as the log files from opmn itself. Always keep a close eye on these since most of the times when things do not look right, these log files will give you good information to what could be the problem.

The other important log file is the log produced by the Sip Servlet Container itself. Under each instance (for example *ps1*), you have the sdp log directory and here you will find the `trace.log` file. Keep a very close eye on this file. You will find this file for the *ocms* instance as well. Make sure you track all of those trace.log files for all instances.

**opmnctl**: The opmnctl tool is used for starting and stopping the server. It also lists the statuses of the instances running. The basic command is:

```
$ORACLE_HOME/opmn/bin/opmnctl status
```

It lists all processes and if they are running or not. Here is typical output:

```
Processes in Instance: priv5.priv5
--------------------
ias-component | process-type | pid | status
--------------------
OC4JGroup:presence | OC4J:ps3 | 4868 | Alive
OC4JGroup:presence | OC4J:ps2 | 4869 | Alive
OC4JGroup:presence | OC4J:ps1 | 4867 | Alive
OC4JGroup:default_group | OC4J:ocms | 4866 | Alive
OC4JGroup:default_group | OC4J:home | 4865 | Alive
ASG | ASG | N/A | Down
```

This output shows that ps1 - ps3 are running on this machine. The ocms instance is also up and running; remember, this is where the User Dispatcher is running. You can also use `opmnctl` to list all the ports on which each instance is listening. Do this by supplying the switch `-l` to opmnctl as in this example:

```
$ORACLE_HOME/opmn/bin/opmnctl status -l
```

This will display output similar to that in Table 5-1. The output in this table has been reduced; normally it includes more information about process id, up time, and other information.

If you have configured everything correctly, you will see that ocms listens at SIP port *5060*, *ps1* at *5062*, and so on.

*Table 5–2 Ports used by process type*

| Process Type | Ports |
|---|---|
| OC4J:ps3 | jms:12605,sip:5066,http:8903,rmis:12705,sip:5066,sips:5067,rmi:12405 |
| OC4J:ps2 | jms:12604,sip:5064,http:8902,rmis:12704,sip:5064,sips:5065,rmi:12404 |
| OC4J:ps1 | jms:12603,sip:5062,http:8901,rmis:12703,sip:5062,sips:5063,rmi:12403 |
| OC4J:ocms | jms:12602,sip:5060,http:7785,rmis:12702,sip:5060,sips:5061,rmi:12402 |
| OC4J:home | jms:12601,http:8888,rmis:12701,rmi:12401 |

## Deploy and Configure Presence

Using Oracle Enterprise Manager, use the group view and deploy the application to all instances in that group. Oracle recommends that you use *presence* as the name of the application during deployment. Once that is done you must configure the following items on all instances:

- Change the `PresRulesXCAPUri` and the `PIDFManipulationXCAPUri`
- Update the `UserAgentFactoryServiceImpl.xml`
- Turn on JGroups

**Change the PresRulesXCAPUri and the PIDFManipulationXCAPUri** For each OC4J Presence Node instance, go to the *Presence* application MBean and set the values of the following attributes:

```
- PIDFManipulationXCAPUri - sip:<xdmsHostIP>;transport=TCP;lr
- PresRulesXCAPUri - sip:<xdmsHostIP>;transport=TCP;lr
```

In our example network, since we have the XDMS pool on the load balancer at *192.168.0.150:5062*, then for all the presence instances *ps1*, *ps2* and *ps3*, the settings will be as follows:

```
PIDFManipulaitonXCAPUri – sip:192.168.0.150:5062;transport=TCP;lr
PresRulesXCAPUri – sip:192.168.0.150:5062;transport=TCP;lr
```

**Update the UserAgentFactoryService Port** For each OC4J Presence Node instance, go to the `UserAgentFactoryService` MBean in the presence application and set the value of the Port attribute to be unique for each of the presence instances on the machine to avoid port conflicts. For ease of management, we recommend that you use consecutive available ports. In our example network, we use the *5070*, *5071* and *5072* for the instances *ps1*, *ps2* and *ps3* respectively.

**Turn on JGroups** For each OC4J Presence Node instance, go to the *PackageManager* MBean and set the values of the following attribute: `JGroupsBroadcastEnabled – true`

Leave the value of `JgroupXMLConfigPath` empty in order to user the default values for JGroups configuration. The default JGgroups configuration uses the following values:

- Multicast Address: *230.0.0.1*
- Multicast Port: *7426*
- Time To Live: *1*

### Configure the User Dispatcher

Configure the User Dispatcher to be able to route SIP traffic to all the presence instances in the deployment. Every User Dispatcher must be configured to direct SIP traffic to all the presence instances on the same machine on which the User Dispatcher is located as well as all the presence instances on the other machines in the deployment. In other words, the User Dispatcher on each presence node (presence machine) must know about all the presence instances on other nodes. To configure the User Dispatcher to route SIP traffic to any presence server, follow these steps:

1. Log into Enterprise Manager on the management node.
2. From the cluster view, select the presence node whose User Dispatcher you want to configure.
3. **Select Applications -> userdispatcher Application Defined Mbeans**.
4. Click **presence-pool** and select **Servers**.
5. Add SIP URIs pointing to all the presence servers in the deployment. The URIs are of the form:

   ```
   sip:<ip-address>:<port>;transport=tcp;lr
   ```

   In the example network, there are three presence nodes (machines), each with three presence server instances, for a total of nine presence servers. Each User Dispatcher must be configured to be able to route SIP traffic to the nine presence servers. We therefore add the following to the presence pool for each of our User Dispatchers:

   ```
   sip:192.168.0.10:5062;transport=tcp;lr
   sip:192.168.0.10:5064;transport=tcp;lr
   sip:192.168.0.10:5066;transport=tcp;lr
   sip:192.168.0.11:5062;transport=tcp;lr
   sip:192.168.0.11:5064;transport=tcp;lr
   sip:192.168.0.11:5066;transport=tcp;lr
   sip:192.168.0.12:5062;transport=tcp;lr
   sip:192.168.0.12:5064;transport=tcp;lr
   sip:192.168.0.12:5066;transport=tcp;lr
   ```

### Tune the Installation

This section contains information that will help you to tune the installed components so that they coexist and run better.

**Turn Off the WebCenter Instance** Once installed, disable the WebCenter instance so that it does not consume resources unnecessarily. Disable it by editing the `opmn.xml` file under `$ORACLE_HOME/opmn/conf/`. Change the status from *enabled* to *disabled* as in the example: `<process-type id="OC4J_WebCenter" module-id="OC4J" status="disabled">`. Restart the server for the changes to take effect:

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

then

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

**Turn off the Home Instance** Now that everything has been installed, turn off the Home OC4J instance on all nodes except the management node. The management node is where you will be able to log into the Enterprise Manager console and view or change the configuration of the whole deployment. To turn off the home instance, edit the

`opmn.xml` file on all the presence nodes and mark the home instance as disabled in the same way you did for the WebCenter instance. The home was only necessary when creating new instances through the `createinstance` command.

## Install the XDM Nodes

Installing the XDM Node is similar to installing a Presence Node. Both of these types of nodes have a User Dispatcher deployed, and they both have the X number of extra instances running with the presence application ear file deployed onto them. The difference is that the XDM nodes also have the AggregationProxy and Subscriber Data Service deployed. Also, they are configured slightly differently when it comes to the User Dispatcher and the XDMS application.

---

**Note:** All nodes should be installed and configured at the same time to ensure smoother deployment.

---

The steps for installing an XDM node are:

1. Install Oracle Application Server 10.1.3.2 on the machine.
2. Apply the 10.1.3.4 patch to the Oracle Application Server.
3. Using the OCMS Installer, install a first instance that will contain the User Dispatcher, Aggregation Proxy and Subscriber Data Service. By default, the Sip Servlet Container will also be installed.
4. Create as many extra OC4J instances as needed.
5. Configure these newly created OC4J instances.
6. Deploy and configure the presence application ear onto those newly created OC4J instances.
7. Configure the User Dispatcher
8. Tune the installation.

---

**Note:** After installing the first XDM node, ensure that subsequent XDM nodes point to the same database as the first.

Also while installing subsequent XDM nodes, you should select the option labeled *Do you wish to reuse the existing schemas*?

---

### Install Oracle Application Server 10.1.3.2

You must first install Oracle Application Server onto the XDMS Node:

1. Start Oracle Universal Installer to install Oracle Application Server 10.1.3.2.
2. Choose **Advanced Installation**.
3. Choose **Oracle WebCenter Framework** as the installation type.
4. Enter a discovery address and ensure that the value you use for the multicast address is the same for all nodes in the cluster. More specifically, the PS and XDMS Nodes must have the same discovery address as the Management Node, otherwise they will not be detected by it. For more information, see your Oracle Application Server installation documentation.

### Apply the 10.1.3.4 Patch to the Oracle Application Server

Next you must apply the Oracle Application Server patch:

1. Apply the 10.1.3.4 WebCenter patch by running Oracle Universal Installer for the 10.1.3.4 patch. For more information, see [title of document for 10.1.3.4 patch].
2. Install Java 5 update 14. The recommended way to do this is to follow these steps:
   - Run the Sun Java installer for JDK 1.5 update 14 to install the JDK to a directory of your choice that refers to `<jdk-directory>`.
   - Go to $ORACLE_HOME and back up the JDK installed there by renaming the file `jdk` to `jdk.install.backup`.
   - Create a symbolic link named `jdk` in $ORACLE_HOME that points to <jdk-directory>.

### Install User Dispatcher

Follow these steps to install User Dispatcher:

1. Using Oracle Universal Installer for OCMS, select only the *Sip Servlet Container*, *User Dispatcher*, *Subscriber Data Services* and the *Aggregation Proxy*.
2. Use the default SIP port value of *5060* for this instance.
3. Complete the installation steps in Oracle Universal Installer.

The Installer creates another OC4J instance named *ocms* executing in your Oracle Application Server. The only applications deployed on this instance are the *User Dispatcher*, *Subscriber Data Services* and the *Aggregation Proxy*.

### Create More OC4J Instances

Use the `createinstance` command in `$ORACLE_HOME/bin` to create more OC4J instances. For instance:

```
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms1 -groupName xdms -httpPort 8901
-defaultAdminPass
```

where *xdms1* is the name of the instance, and the group is named *xdms*. If the group did not exist, then it will be created.

*-httpPort 8901* specifies the port number, and Oracle recommends using consecutive available ports for ease of management. In our example network, we use *8901*, 8902 and 8903 for the OC4J instance *xdms1*, *xdms2* and *xdms3* respectively.

*-defaultAdminPass* omits setting a password at this time. This directs `createinstance` to omit setting a password for the instance. Later you will issue another command to set the password for the instance. If you do not include this option, `createinstance` will prompt you for a password. If you are creating just a few instances, you can do it this way, but if you want to execute a script for instance creation, it is more efficient to use *-defaultAdminPass*.

To set the password for the instance you just created, execute this command:

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../<instance-name> -jar jazn.jar -activateadmin
<password>.
```

For example, to set the password for the *xdms* instance to *myPassword1*, you would execute:

```
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms1 -jar jazn.jar -activateadmin myPassword1
```

Repeat these two commands as many times as needed in order to create enough instances. In our example network, we need to run the pair of commands three times, to create three OC4J instances *xdms1*, *xdms2* and *xdms3* for the three XDMS instances respectively:

```
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms1 -groupName xdms -httpPort 8901
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms1 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms2 -groupName xdms -httpPort 8902
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms2 -jar jazn.jar -activateadmin myPassword1
cd $ORACLE_HOME/bin
./createinstance -instanceName xdms3 -groupName xdms -httpPort 8903
-defaultAdminPass
cd $ORACLE_HOME/j2ee/home
java -Doracle.j2ee.home=../xdms3 -jar jazn.jar -activateadmin myPassword1
```

### Configure OC4J Instances

The previous step outlined how to create new OC4J instances to which the actual XDMS Server will be deployed. Before you can deploy, you must configure those new instances to also pick up the Sip Servlet Container. The necessary jars have already been installed into the Oracle Application Server when you installed the User Dispatcher, so all you need to do is to further configure each of those *xdmsX* instances to pick up the Sip Servlet Container jars from the shared library. You must also configure the new OC4J instances for proper startup and logging. These are the overall steps you must perform for all the new OC4J instances:

1. Configure the instances to pick up the shared library where all the necessary jars for the Sip Servlet Container reside. This involves editing the `boot.xml` file.
2. Configure logging. For your new instances to use the logging done by the XDMS instance, you must edit the `j2ee-logging.xml` file.
3. Edit the start-up and shut-down parameters for the instances so that the Sip Servlet Container is loaded. This is done by editing the `$ORACLE_HOME/opmn.xml` file.
4. Specify the SIP ports to which the new instances should be listening.
5. Configure the Sip Servlet Container to listen to the correct IP address.
6. Add the `xcap config` directory.
7. Verify your configuration.

**Shut down the instance** `$ORACLE_HOME/opmn/bin/opmnctl stopall`

**Configure the instance to use the shared libraries** Configure the instances to pick up the shared library where necessary jars for the Sip Servlet Container reside. To achieve this, copy the `boot.xml` from the *ocms* instance that was created by the OCMS installer:

```
cp $ORACLE_HOME/j2ee/ocms/config/boot.xml $ORACLE_HOME/j2ee/<instance
name>/config/
```

This copies the correctly-configured boot.xml found in the ocms instance into another instance. Repeat this for all the newly created instances. In our example network, we issue the above command three times, replacing <*instance name*> with *xdms1*, *xdms2* and *xdms3* respectively to copy boot.xml into the three OC4J instances.

**Configure logging** Just as you copied the boot.xml file to load the shared libraries you can copy the configuration file for logging. Copy the `j2ee-logging.xml` file found in the config directory of ocms over to all your instances:

```
cp $ORACLE_HOME/j2ee/ocms/config/j2ee-logging.xml $ORACLE_HOME/j2ee/<instance
name>/config/
```

In our example network, we issue the above command three times, replacing <instance name> with *xdms1*, *xdms2* and *xdms3* respectively to copy `j2ee-logging.xml` into the three OC4J instances.

**Configure JVM start/stop parameters** Configure the JVM parameters to set the start-up and shut-down of the instances so that the Sip Servlet Container is loaded. This is done by editing the `opmn.xml` file to set the start and stop parameters on all the presence OC4J instances (*xdms1*, *xdms2* and *xdms3* in our example network) as well as the ocms instance.

**Start parameters:**

```
<data id="java-options" value="-server -Xmx2500M -Xms2500M
-Xloggc:/home/sdp/logs/gc.xdms1.log -XX:+PrintGCDetails -XX:NewRatio=3
-XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -Dhttp.maxFileInfoCacheEntries=-1 -Djava.security.policy=$ORACLE_
HOME/j2ee/home/config/java2.policy -Djava.awt.headless=true
-Dhttp.webdir.enable=false -DopmnPingInterval=1
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.s
ipservletcontainer.deployer.Oc4jApplicationHook "/>
```

**Stop parameters:**

```
<data id="java-options" value="-Djava.security.policy=$ORACLE_
HOME/j2ee/home/config/java2.policy -Djava.awt.headless=true
-Dhttp.webdir.enable=false
-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.s
ipservletcontainer.deployer.Oc4jApplicationHook"/>
```

The meaning of these parameters is explained below:

-*Xmx2500M* Set the maximum JVM memory to 2.5GB

-*Xms2500M* Set the minimum JVM memory to 2.5GB

-*XX:+PrintGCDetails* Enable logging of collection activity

-*XX:NewRatio=3* Set the ratio between the young generation and the old generation of the heap to 1:3 (in other words, the combined sizes of the eden space plus survivor space is one fourth of the total heap size).

-*XX:+UseConcMarkSweepGC* Enable the concurrent mark-and-sweep garbage collector (also known as the concurrent low pause collector).

-*XX:+UseParNewGC* use parallel threads.

-*XX:PermSize=128m* Set the initial size of the permanent generation to 128MB.

-*XX:MaxPermSize=128m* Set the maximum size of the permanent generation to 128MB.

*-Xss128k* Set the stat size for each thread to 128 KB.

*-Dhttp.maxFileCacheEntries=-1* Disable caching on the HTTP server that is bundled with the Oracle Application Server. In this release, caching must be disabled in order to achieve good performance for the Presence Server.

*-Doracle.hooks=oracle.sdp.sipservletcontainer.SipServletContainerOc4j;oracle.sdp.sipservletcontainer.deployer.Oc4jApplicationHook* Enables OC4J to load the SipContainer.

---

**Note:** The ocms instance you installed does not contain all parameters needed for a multiple JVM installation, so make sure that the above parameters are set on the ocms instance as well.

---

**Configure the SIP Ports** Now that you have set the start and stop parameters, you must configure the Sip Servlet Container to listen on the correct ports. By default, the Sip Servlet Container will listen to port *5060* for SIP and *5061* for SIPS. If you look at the file `$ORACLE_HOME/opmn/conf/opmn.xml`, you will see that ocms instance that was created by the OCMS installer is configured with these default values as shown below:

```
<port id="sip" range="5060"/>
<port id="sips" range="5061"/>
```

Those values for the ocms instance should remain as-is, since you want the User Dispatcher to listen for incoming traffic on those ports. For the other OC4J instances that you created for presence nodes, you must configure different ports for each of them to avoid port conflicts. For ease of management, it is recommended to use available ports in series. In our example network, we assign the ports for the presence instances beginning with *5062* – yielding the following configuration:

*ps1*

```
<port id="sip" range="5062"/>
<port id="sips" range="5063"/>
```

*ps2*

```
<port id="sip" range="5064"/>
<port id="sips" range="5065"/>
```

*ps3*

```
<port id="sip" range="5066"/>
<port id="sips" range="5067"/>
```

**Configure the Sip Servlet Container to listen ton the correct IP address** By default the Sip Servlet Container will listen on IP address *127.0.0.1*. You must change that to be the externally-addressable IP address of the machine. In the previous steps you configured the Oracle Application Server to load up the container but do not have any configuration files for the actual Sip Servlet Container itself. In order to generate the default configuration files, you must start the Oracle Application Server and then stop it again. Once the default configuration files are generated, you can edit them to suit your deployment.

To start the Oracle Application Server:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

Once it is up and running, shut it down:

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

A new configuration directory named *sdp* now exists under `$ORACLE_HOME/j2ee/<instance name>/config/sdp/`. Here you will find all the information related to configuration of OCMS on the respective instances. Each instance has a file named `SipServletContainer.xml`. You must edit this file for each of the instances and change the IPAddress attribute from 127.0.0.1 to the IP Address at which your Sip Servlet Container should listen. However, since this file should be identical for all Sip Servlet Containers on this machine, you do not have to edit each one manually – instead, copy the one found under the *ocms* directory and reuse it. Here is the command:

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/SipServletContainer.xml $ORACLE_
HOME/j2ee/<instance name>/config/sdp/
```

Replace <*instance-name*> with the name of the presence OC4J instances. In our example network, we execute the above command three times, once each for *xdms1*, *xdms2* and *xdms3*. Now that the Sip Servlet Container is configured for all the instances, you can start the server. To start all the OC4J instances on the Oracle Application Server:

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

**Configure the xcap config directory** Even though the `ua-profile` event package is not explicitly used on the presence nodes, it must be configured to be present; therefore we need to you must copy the xcap configuration files to all the presence OC4J instances. To do so, copy the xcap directory into the sdp config directory `$ORACLE_HOME/j2ee/<instance-name>/config/sdp/` for each instance. In our example network, we must copy the xcap directory into:

```
$ORACLE_HOME/j2ee/xdms1/config/sdp/
$ORACLE_HOME/j2ee/xdms2/config/sdp/
$ORACLE_HOME/j2ee/xdms3/config/sdp/
```

You will find the xcap directory under the directory where you extracted the OCMS installer. If, for instance, you extracted the installer into the directory `OCMS_INSTALLER`, then the xcap directory will be under:

```
OCMS_
INSTALLER/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/DataFiles/Expanded/Shipho
me/shiphome-archive/xcapconf/conf
```

**Configure Aggregation Proxy** You must configure the Aggregation Proxy on each XDMS node to point to the User Dispatcher on the same node. Follow these steps to configure the Aggregation Proxy:

1. Log onto the Oracle Enterprise Manager console on the management node and and go to the *Aggregation Proxy Mbean*.
2. Modify `XCAPRoot` to be the context root of the User Dispatcher on the same node; the default value of the User Dispatcher context root is *userdispatcher*, so unless you changed it, set the value of the XCAPRoot attribute to */userdispatcher*. You do not need to change the other attributes such as *XCAPHost* and *XCAPPort* since these should be set correctly by the Installer.

**Configure springbeans.xml** Copy the `springbeans.xml` from the sdp config directory of the ocms instance into the sdp config directory of all the presence instances:

```
cp $ORACLE_HOME/j2ee/ocms/config/sdp/springbeans.xml $ORACLE_HOME/j2ee/<instance
name>/config/sdp/
```

In our example network, we execute the above command three times, replacing <instance-name> with *xdms1*, *xdms3* and *xdms3* respectively.

**Verify your configuration settings** Before continuing with the installation, verify that your actions were successful by examining the logs. You can also issue the `netstat` command to verify that the server is listening on the correct ports.

**Log Files:** You should also regularly monitor the log files; they will reveal if anything is wrong with the configuration. There are two main log files you should pay attention to. The first one is the console output from each of the instances, which is located at `$ORACLE_HOME/opmn/logs` and a typical file will be named like so for the xdms nodes:

```
xdms~<instance-name>~default_group~1.log
```

You will also find the one for the ocms instance as well as the log files from opmn itself. Always keep a close eye on these since most of the times when things do not look right, these log files will give you good information to what could be the problem.

The other important log file is the log produced by the Sip Servlet Container itself. Under each instance (for example *xdms1*), you have the sdp log directory and here you will find the `trace.log` file. Keep a very close eye on this file. You will find this file for the *ocms* instance as well. Make sure you track all of those trace.log files for all instances.

**opmnctl**: The opmnctl tool is used for starting and stopping the server. It also lists the statuses of the instances running. The basic command is:

```
$ORACLE_HOME/opmn/bin/opmnctl status
```

It lists all processes and if they are running or not. Here is typical output:

```
Processes in Instance: priv5.priv5
--------------------
ias-component | process-type | pid | status
--------------------
OC4JGroup:xdms | OC4J:xdms3 | 4868 | Alive
OC4JGroup:xdms | OC4J:xdms2 | 4869 | Alive
OC4JGroup:xdms | OC4J:xdms1 | 4867 | Alive
OC4JGroup:default_group | OC4J:ocms | 4866 | Alive
OC4JGroup:default_group | OC4J:home | 4865 | Alive
ASG | ASG | N/A | Down
```

This output shows that xdms1 - xdms3 are running on this machine. The ocms instance is also up and running; remember, this is where the User Dispatcher is running. You can also use `opmnctl` to list all the ports on which each instance is listening. Do this by supplying the switch `-l` to opmnctl as in this example:

```
$ORACLE_HOME/opmn/bin/opmnctl status -l
```

This will display output similar to that in Table 5-1. The output in this table has been reduced; normally it includes more information about process id, up time, and other information.

If you have configured everything correctly, you will see that ocms listens at SIP port *5060*, *xdms1* at *5062*, and so on.

*Table 5–3 Ports used by process type*

| Process Type | Ports |
|---|---|
| OC4J:xdms3 | jms:12605,sip:5066,http:8903,rmis:12705,sip:5066,sips:5067,rmi:12405 |
| OC4J:xdms2 | jms:12604,sip:5064,http:8902,rmis:12704,sip:5064,sips:5065,rmi:12404 |
| OC4J:xdms1 | jms:12603,sip:5062,http:8901,rmis:12703,sip:5062,sips:5063,rmi:12403 |
| OC4J:ocms | jms:12602,sip:5060,http:7785,rmis:12702,sip:5060,sips:5061,rmi:12402 |
| OC4J:home | jms:12601,http:8888,rmis:12701,rmi:12401 |

**Create Connection Pool** Now create the *Connection Pool* and the *Data Sources* for the XDMS instances. Remember that the user presence rules and PIDF documents are stored on an Oracle Database (located at *192.168.0.30* in our example network), and the XDMS nodes must be configured to be able to access these documents. Set up a connection pool and data source with the following properties:

- Connection Pool Name: `SDP XDMS Oracle Connection Pool`
- Connection Factory Class: `oracle.jdbc.pool.OracleDataSource`
- URL: jdbc:oracle:thin: `@//<db-hostname>:<db-port>/<db-name>`
- OC4J Username: `oc4jadmin`
- OC4J password: `myPassword1`
- Database Username: `SDP_ORASDPXDMS`
- Database Password: `myDBPassword1`
- Data Source Name: `OcmsXdmsDs`

You can change the username and password to suit your deployment, as long as those credentials are valid for accessing the Oracle Database. To create the connection pool and the data source, you must execute the following commands on the management node:

```
java -jar ORACLE_HOME/j2ee/home/admin_client.jar
deployer:clusterj:opmn://<management-host-ip-address> <oc4j-username> <admin_pwd>
-addDataSourceConnectionPool -applicationName <name> -name <connection pool name>
-name <connection pool name> -factoryClass <factory class> -dbPassword <db
password> -dbUser <db username> -url <url>

java -jar ORACLE_HOME/j2ee/home/admin_client.jar
deployer:cluster:opmn://<management-host-ip-address>/xdms <oc4j-username>
<password> -addManagedDataSource -applicationName default -name
"<data-source-name>" -jndiLocation java:jdbc/<data-source-name>
-connectionPoolName "<connection-pool-name>"
```

In our example network, the Oracle Database is located on host *192.168.0.30* on port *1521*,the database name is orcl11g and our management node is located on *192.168.0.100*. Given that example network and the configuration above, we will execute the commands below on the management node to create the JDBC data sources for the XDMS:

```
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -addDataSourceConnectionPool -applicationName default -name "SDP XDMS
Oracle Connection Pool" -factoryClass oracle.jdbc.pool.OracleDataSource -dbUser
SDP_ORASDPXDMS -dbPassword myDBPassword1 -url
jdbc:oracle:thin:@//192.168.0.30:1521/orcl11g
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
```

```
myPassword1 -addManagedDataSource -applicationName default -name "OcmsXdmsDs"
-jndiLocation java:jdbc/OcmsXdmsDs -connectionPoolName "SDP XDMS Oracle Connection
Pool"
```

If you need to remove the Data Source and/or the Connection Pool later, you can issue the following commands:

```
java -jar admin_client.jar
deployer:cluster:opmn://<management-host-ip-address>/xdms <oc4j-username>
<oc4j-password> -removeManagedDataSource -name "<data-source-name>"
java -jar admin_client.jar
deployer:cluster:opmn://<management-host-ip-address>/xdms <oc4j-username>
<oc4j-password> -removeDataSourceConnectionPool -name "<connection-pool-name>"
```

In our example network, the corresponding command to remove the data source and connection pools respectively would be:

```
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -removeManagedDataSource -name "OcmsXdmsDs"
java -jar admin_client.jar deployer:cluster:opmn://192.168.0.100/xdms oc4jadmin
myPassword1 -removeDataSourceConnectionPool -name "SDP XDMS OracleConnection Pool"
```

### Deploy and Configure XDMS

Using Oracle Enterprise Manager, use the group view and deploy the presence application .ear file (use the same .ear file for both Presence and XDMS) to all instances in that group. Oracle recommends that you use *xdms* as the name of the application during deployment. Once that is done you must configure the following items on all instances:

- Change the `PublicXCAPRootUrl` and the `PublicContentServerRootUrl`
- Update the `UserAgentFactoryServiceImpl.xml`
- Turn on JGroups

**Change the PublicXCAPRootUrl and the PublicContentServerRootUrl** For each OC4J XDMS Node instance, go to the *XDMS -> XCAPConfig* application MBean and set the values of the following attributes:

```
- PublicXCAPRootUrl - http://<node-ip>:<node-http-port>/services
- PublicContentServerRootUrl - http://<node-ip>:<node-http-port>/contentserver
```

Where *node-ip* is the IP address of the machine on which the XDMS instance resides, and *node-http-port* is the HTTP port of the OC4J instance on which the XDMS server is running; this is the value supplied using the `-httpPort XXXX` option when creating the OC4J instances using the `createinstance` command.

In our example network, we have three XDMS instances per XDMS node; therefore we set the values as follows for the XDMS instances on host *192.168.0.20*:

**xdms1:**

```
PublicXCAPRootUrl - http://192.168.0.20:8091/sevices
PublicContentServerRootUrl - http://192.168.0.20:8091/contentserver
```

**xdms2:**

```
PublicXCAPRootUrl - http://192.168.0.20:8092/sevices
PublicContentServerRootUrl - http://192.168.0.20:8092/contentserver
```

**xdms3:**

```
PublicXCAPRootUrl - http://192.168.0.20:8093/sevices
PublicContentServerRootUrl - http://192.168.0.20:8093/contentserver
```

The settings on the second XDMS node/machine of our example network at *192.168.0.21* would be similar to the above, with the only difference being that the IP address in the attributes would be *192.168.0.21* instead of *192.168.0.20*.

**Update the UserAgentFactoryService Port** For each OC4J XDMS Node instance, go to the `UserAgentFactoryService` MBean in the XDMS application and set the value of the *Port* attribute to be unique for each of the XDMS instances on the machine to avoid port conflicts. For ease of management, we recommend that you use consecutive available ports. In our example network, we use the *5070*, *5071* and *5072* for the instances *xdms1*, *xdms2* and *xdms3* respectively.

**Turn on JGroups** For each OC4J XDMS Node instance, go to the *PackageManager* MBean and set the values of the following attribute: JGroupsBroadcastEnabled – true

Ensure that the XDMS servers and the Presence servers are not listening on the same port and address for JGroups notification, otherwise errors will occur. Since the default JGroups configuration for the Presence servers was used, a different configuration for the XDMS servers must be used. To do so, supply the path to the JGroups configuration file to be read by the XDMS server package manager. Here is an example of a minimal configuration file:

```
<config>
  <UDP bind_addr="[host-ip-address]" mcast_addr="[multicast-address]" mcast_
port="[multicast-port]"  ip_ttl="1"/>
</config>


replacing the variables as follows:

[host-ip-address] - the IP address of the host running the XDMS server

[multicast-address] - the multicast address on which all participants in the group
will listen for messages.

[multicast-port] - the multicast port on which all participants in the group will
listen for messages.
```

Ensure that all the XDMS server instances on all the nodes have the same value for *multicast-address* and *multicast-port*; XDMS servers on the same host will usually also have the same *host-ip-address*. However, the combination of *multicast-address:multicast-port* for the XDMS servers must be different from the one for the Presence servers. Remember that for the Presence servers, the default configuration value (*230.0.0.1:7426*) was used for *multicast-host:multicast-port*. Save the above file to a location of your choosing, and then edit the `JGroupXMLConfigPath` attribute of the *EventPackageManager* MBean to point to this file:

```
JGroupXMLConfigPath - <absolute path to the jgroups XML configuration file>
```

In our example network, we use the following jgroups configuration file on the host at *192.168.0.20*:

```
<config>
  <UDP bind_addr="192.168.0.20" mcast_addr="234.0.0.1" mcast_port="1234"  ip_
ttl="1"/>
</config>
```

All the XDMS instances on this node must share the same configuration, so save this file into `$ORACLE_HOME/j2ee/ocms/config/sdp/jgroups.xml` and then for each of the XDMS instances *xdms1*, *xdms2* and *xdms3*, edit the *EventPackageManager* MBean with the following settings:

```
JGroupBroadcastEnabled - true
JGroupXMLConfigPath - <path-to-ORACLE_HOME>/j2ee/ocms/config/sdp/jgroups.xml
```

Configure the XDMS host at *192.168.0.21* in the same way, replacing the host IP address as appropriate.

**Database Configuration** Complete these configuration steps:

1. Copy `orasdpxdms.create.oracle.sql` and `xcapservice.create.oracle.sql` files to the machine where database is running. These files can be found at: `<installer files extraction location>/Disk1/stage/Components/oracle.sdp/10.1.3.4.0/1/Data Files/Expanded/DBFiles/`
2. Log in to the machine where the database is running; ensure you are in the same directory where you copied the above sql files.
3. Connect to the database as *sysdba* using sqlplus. This can be done as `bash$ sqlplus  / as sysdba`
4. Run the above sql files as follows at the sqlplus prompt:

   sqlplus> @orasdpxdms.create.oracle.sql PREFIX  DATADIR  PASSWORD

   where PREFIX is what you will use as a prefix for your schemas and users, DATADIR is the path to where the created database files will reside, and PASSWORD is the password for the users which will be created.

   for example:

   sqlplus> @orasdpxdms.create.oracle.sql  TEST  "C:\oraexe\oradata\XE" myPassword1

   will create the data base files under C:\oraexe\oradata\XE. The schema names and user names will start with TEST and the password for the users will be myPassword1.
5. Run the other sql file as follows:

   sqlplus> @xcapservice.create.oracle.sql PREFIX

   where PREFIX should be the same as given while executing the `orasdpxdms.create.oracle.sql` file.

**Sash Configuration** Complete these configuration steps:

1. Go to `$ORACLE_HOME/sdp/sash/sbin`
2. Create a file: `xdms-create-default-appusage.txt`
3. Add the following lines to the file:

```
xcap appusage create applicationUsage=pres-rules
configurationFilename=presrules_au.xml

xcap appusage create applicationUsage=resource-lists
configurationFilename=resource-lists_au.xml

xcap appusage create applicationUsage=pidf-manipulation
```

```
configurationFilename=pidfmanipulation_au.xml
```

4. To seed the database with default application usages, execute the following:

```
- bash$ sash -a presenceapplication --username oc4jadmin --password PASSWORD
--file xdms-create-default-appusage.txt
```

   where PASSWORD is the password for oc4jadmin.

**Verification** - Log on to sash as follows:

bash$ sash -a presenceapplication --username oc4jadmin --password PASSWORD

At the sash prompt enter:

sash# xcap appusage list

If you configured correctly, this will return three values: `resource-lists`, `pidf-manipulation`, and `pres-rules`

### Configure the User Dispatcher

Configure the User Dispatcher to be able to route SIP traffic to all the XDMS instances in the deployment. Every User Dispatcher must be configured to direct SIP traffic to all the presence instances on the same machine on which the User Dispatcher is located as well as all the XDMS instances on the other machines in the deployment. In other words, the User Dispatcher on each XDMS node (XDMS machine) must know about all the XDMS instances on other nodes. To configure the User Dispatcher to route SIP traffic to any XDMS server, follow these steps:

1. Log into Enterprise Manager on the management node.
2. From the cluster view, select the XDMS node whose User Dispatcher you want to configure.
3. **Select Applications -> userdispatcher Application Defined Mbeans**.
4. Click **xdms-sip-pool** and select **Servers**.
5. Add SIP URIs pointing to all the XDMS servers in the deployment. The URIs are of the form:

```
sip:<ip-address>:<port>;transport=tcp;lr
```

   In the example network, there are two XDMS nodes (machines), each with three XDMS server instances, for a total of six XDMS servers. Each User Dispatcher must be configured to be able to route SIP traffic to the six XDMS servers. We therefore add the following to the xdms sip pool for each of our User Dispatchers:

```
sip:192.168.0.20:5062;transport=tcp;lr
sip:192.168.0.20:5064;transport=tcp;lr
sip:192.168.0.20:5066;transport=tcp;lr
sip:192.168.0.21:5062;transport=tcp;lr
sip:192.168.0.21:5064;transport=tcp;lr
sip:192.168.0.21:5066;transport=tcp;lr
```

6. Click `xdms-http-pool` and select **Servers**.
7. Add HTTP URIs pointing to all the XDMS instances in the deployment. The URIs are of the form:

```
http://<ip-address>:<port>/services
```

In the example network, there are two XDMS nodes (machines), each with three XDMS server instances, for a total of six XDMS servers. Each User Dispatcher must be configured to be able to route HTTP traffic to the six XDMS servers. We therefore add the following to the xdms sip pool for each of our User Dispatchers:

```
http://192.168.0.20:8901/services/
http://192.168.0.20:8902/services/
http://192.168.0.20:8903/services/
http://192.168.0.21:8901/services/
http://192.168.0.21:8902/services/
http://192.168.0.21:8903/services/
```

### Tune the Installation

This section contains information that will help you to tune the installed components so that they coexist and run better.

**Update Overload Policy** For all the Presence and XDMS instances except the User Dispatchers, modify the `SipSessionTableMaxSize attribute` to 400000 in `OverloadPolicy.xml`. `OverloadPolicy.xml` is in `$ORACLE_HOME/j2ee/<instance-name>/config/sdp`.

**Turn Off the WebCenter Instance** Once installed, disable the WebCenter instance so that it does not consume resources unnecessarily. Disable it by editing the `opmn.xml` file under `$ORACLE_HOME/opmn/conf/`. Change the status from *enabled* to *disabled* as in the example: `<process-type id="OC4J_WebCenter" module-id="OC4J" status="disabled">`. Restart the server for the changes to take effect:

```
$ORACLE_HOME/opmn/bin/opmnctl stopall
```

then

```
$ORACLE_HOME/opmn/bin/opmnctl startall
```

**Turn off the Home Instance** Now that everything has been installed, turn off the Home OC4J instance on all nodes except the management node. The management node is where you will be able to log into the Enterprise Manager console and view or change the configuration of the whole deployment. To turn off the home instance, edit the `opmn.xml` file on all the presence nodes and mark the home instance as disabled in the same way you did for the WebCenter instance. The home was only necessary when creating new instances through the `createinstance` command.

**Turn Off ASG Instance** Turn off the ASG instance to enhance performance.

## Configure the Load Balancer

From an outside perspective, the entire network will appear as one node and this is achieved by having one or more Load Balancers in front of the Presence and XDM Cluster. This section describes how to setup the BigIP Load Balancer from F5 and will use the example network as previously described.

From a high-level perspective you will need to create two pools for the SIP traffic and one pool for the XCAP traffic. One SIP pool will contain all the Presence Nodes in the system and the other will contain all the XDM Nodes. The pool for the XCAP traffic will only contain a list of the XDM Nodes since those nodes are the only ones dealing with XCAP traffic.

An external client will not connect directly to these pools but rather a Virtual Server; this Virtual Server will then contain a particular pool. It is this Virtual Server that the external clients will see and it is what makes the entire system appear as a single box.

All configuration of the BigIP is done through its web-based management interface. See your BigIP documentation for a complete description of its capabilities and configuration options.

### Create New Pools

In order to create a new pool, navigate to the Pool page by choosing **Local Traffic -> Virtual Servers -> Pools**. The general steps for creating a new pool are as follows:

1. Leave the configuration at *Basic* and start off by choosing an appropriate name for the pool.
2. For health monitors, pick the *gateway_icmp*.
3. *Round Robin* is the Load Balancing Method that suits our needs. Choose it.
4. *Priority Group Activation* - Leave disabled.
5. *New Members*: add all nodes that should belong to this particular pool. For each node, enter its address and port, then click **Add**. Repeat this for all the nodes that should go into this particular pool.

To use our example network, the following information would be entered to create our first pool: the *Presence SIP Pool*.

1. The name can be anything, and in this particular example we will use *ps_sip* as the name of this pool.
2. For health monitors, pick the *gateway_icmp*.
3. *Round Robin* is the Load Balancing Method that suits our needs. Choose it.
4. *Priority Group Activation* - Leave disabled.
5. The members to add to this pool are all our Presence Nodes. In the example network, there are three presence nodes and our pool must point to those three instances. Remember, the User Dispatcher is the one front-facing the actual Presence Instances so it is the User Dispatcher that we really are pointing the Load Balancer to. As such, the following three addresses will be added as members to this pool:
   - IP Address: *192.168.0.10 Port: 5060*
   - IP Address: *192.168.0.11 Port: 5060*
   - IP Address: *192.168.0.12 Port: 5060*

   We have now created the presence pool and in the very same way we would create another pool for the SIP traffic going to the XDM Network:

   1. Name: *xdms_sip*
   2. Same as above.
   3. Same as above.
   4. Same as above.
   5. The member will be the two XDMS Nodes in our example network. Just as in the case of the Presence Pool, we will point this pool to the User Dispatchers running on those XDMS Nodes. Therefore, the following two members are added to this node:

IP Address: 192.168.0.20 Port: 5060

IP Address: 192.168.0.21 Port: 5060

The pool for the XCAP traffic is the same as the other two; it just happens to dispatch HTTP traffic instead of SIP traffic (remember that XCAP goes over HTTP) and the concept is the same.

1. Name: *xdms_http*
2. Same as above.
3. Same as above.
4. Same as above.
5. The members will still be the two XDMS nodes but remember that the XCAP traffic must be authenticated and therefore this traffic must go through the Aggregation Proxy. As such, we will not point these members to the User Dispatcher but rather to the HTTP port where the Aggregation Proxy is listening. In our example network, the following addresses would be added:

   IP Address: 192.168.0.20 Port: 80

   IP Address: 192.168.0.21 Port: 80

We have now created the three pools necessary for our network. The next step is to configure the Virtual Servers that will front-face each one of these pools.

### Create New Virtual Servers

A Virtual Server (VS) is what the external clients interact with. When a client connects to a particular VS, the VS will proxy that request to one of its pools, which in turn will dispatch it to one of its members. In our case, we will create one VS front-facing each one of the three pools and the general steps for creating a new VS are listed below. You will find the Virtual Servers under the *Local Traffic*.

1. Click **Create** to start creating a Virtual Server.
2. This VS will be using TCP, so an appropriate name would be *ps_sip_tcp*.
3. In our example network, the load balancer only has one interface (at least only one enabled and we do not need more) and that one is listening on *192.168.0.150* so this is the address we will enter as the destination address.
4. The port can be any available port, but for this example we will use *5060*.
5. This VS is running TCP, so choose that.
6. Enable **Auto Map for the SNAT Pool**.
7. The *Resource* will be the pool containing our Presence Nodes. We named this pool ps_sip, and it will be our default pool. This is the only pool this VS will use.
8. Click **Finish** to create the new Virtual Server.

The next VS to create is the one for SIP traffic going to the XDM Cluster:

1. Click **Create** to start creating a Virtual Server.
2. Name it *xdms_sip_tcp*.
3. Destination address is *192.168.0.150*.
4. Since *5060* is in use, pick *5062*.
5. This VS is running TCP, so choose that.

6. Enable **Auto Map for the SNAT Pool**.
7. The *Resource* will be the pool named *xdms_sip*.
8. Click **Finish**.

A Virtual Server (VS) is what the external clients interact with. When a client connects to a particular VS, the VS will proxy that request to one of its pools, which in turn will dispatch it to one of its members. In our case, we will create one VS front-facing each one of the three pools and the general steps for creating a new VS are listed below. You will find the Virtual Servers under the *Local Traffic*.

1. Click **Create** to start creating a Virtual Server.
2. Name it *xdms_http_tcp*.
3. Destination address is *192.168.0.150*.
4. Port *80*.
5. Protocol is *TCP*.
6. Enable **Auto Map for the SNAT Pool**.
7. The *Resource* will be the pool named *xdms_http*.
8. Click **Finish**.

That concludes the configuration of the F5 BipIP load balancer. See the BigIP documentation for more detailed explanation of the various options.

# 6

# Post-Installation

This chapter describes typical installation and configuration problems and their solution. It contains the following sections:



## Perform Post-Installation Administrative Tasks

You have installed OCMS and verified the installation by using Oracle Communicator to connect to the OCMS SIP Server Container. The OCMS installation is now complete. However, before applications can be deployed on OCMS, the OCMS administrator can perform the following administrative tasks:

- Configure the SIP Servlet Container and the Application Router. Refer to "Configuring the SIP Servlet Container" in Oracle Communication and Mobility Server Administrator's Guide.
- Perform required configuration for Presence, Proxy-Registrar, Edge Proxy, and Aggregation Proxy if these applications were installed. Refer to "Configuring SIP Applications" in the "Managing SIP Applications" chapter in Oracle Communication and Mobility Server Administrator's Guide.
- Provision OCMS users to the Oracle database using the Sapphire Shell (Sash) command-line utility. Refer to the "Provisioning Users and Applications" chapter in Oracle Communication and Mobility Server Administrator's Guide.
- Configure how authorization and authentication are performed for SIP applications and how security is set for SIP servlets. Refer to the "OCMS Security" chapter in Oracle Communication and Mobility Server Administrator's Guide.
- Configure logging for OCMS. Refer to the "Logging" chapter in Oracle Communication and Mobility Server Administrator's Guide.
- When installing the OCMS in a production environment, consider using at least two network cards so that the SIP load traffic and the synchronization data may be separated and not interfere with one another.
- Please refer to your database documentation for more information on database tuning.

# Tune Database

No special database tuning is required to run Oracle Communication and Mobility Server out-of-the-box. However, before a production deployment of OCMS, Oracle recommends you do the following:

- Increase the number of connections from OCMS to the database. For a typical installation, OCMS creates three schema users: one for subscriberdata services (`<prefix>_oraspdsds`), one for location service (`<prefix>_orasdpls`), and one for XDMS (<prefix>_orasdpxdms). Out-of-the-box, both the minimum and maximum number of database connections are configured to be four to each of these schemas.
- Increase the number of processes on the database server. This is especially important if you increase the number of database connections from OCMS as described above, and when there are other applications connecting to the database as well.
- Increase the SGA and PGA cache sizes. Increasing the PGA and SGA sizes allow for a larger amount of data to be stored in cache.
- Add additional database files to each schema if you run out of table space. Out-of-the-box, there is only one database file created for each schema configured with a maximum size of 250MB.
- Please refer to your database documentation for more information on how to tune your database.

# Index

## P



## R



## S



## T



## U



## V



## X