Oracle® Communications WebRTC Session Controller

Installation Guide

Release 7.0

E40974-02

June 2014

Oracle Communications WebRTC Session Controller Installation Guide, Release 7.0

E40974-02

Contents

5 Installing WebRTC Session Controller Signaling Engine

6 Creating and Configuring a WebRTC Session Controller Signaling Engine Domain

7 WebRTC Session Controller Signaling Engine Post-Installation Tasks

8 Installing WebRTC Session Controller Media Engine

9 WebRTC Session Controller Media Engine Post-Installation Tasks

Preface

This guide describes the system requirements and procedures for installing Oracle Communications WebRTC Session Controller.

Audience

This document is for system administrators who install and configure the WebRTC Session Controller. The person installing the software should be familiar with the following topics:

- Operating system commands
- Network Management
- Oracle Coherence

Before reading this guide, you should have a familiarity with WebRTC Session Controller. See *WebRTC Session Controller Concepts*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Accessing Oracle Communications Documentation

WebRTC Session Controller documentation is available from the Oracle Documentation Web site: http://docs.oracle.com.

Related Documents

Refer to these additional documents for related information on WebRTC Session Controller:

- *Oracle Communications WebRTC Session Controller Concepts*
- *Oracle Communications WebRTC Session Controller Extension Developer's Guide*

- *Oracle Communications WebRTC Session Controller System Administrator's Guide*
- *Oracle Communications WebRTC Session Controller Security Guide*
- *Oracle Fusion Middleware 12c Documentation Library*
- *Oracle Communications Application Session Controller System Installation and Commissioning Guide*
- *Oracle Communications Application Session Controller System Administration Guide*
- *Oracle Communications Application Session Controller Objects and Properties Reference Guide*
- *Oracle Communications Application Session Controller System Operations and Troubleshooting*

1 WebRTC Session Controller Installation Overview

This chapter provides an overview of Oracle Communications WebRTC Session Controller installed components and of the WebRTC Session Controller installation process. Subsequent chapters describe installation steps in detail.

For more detailed WebRTC Session Controller overview information, see the discussion about system architecture in *WebRTC Session Controller Concepts*.

About WebRTC Session Controller Media Engine and Signaling Engine

WebRTC Session Controller comprises two main sub components:

- WebRTC Session Controller Signaling Engine (Signaling Engine)
- WebRTC Session Controller Media Engine (Media Engine)

Signaling Engine handles signaling services between Web browser clients, traditional Session Initialization Protocol (SIP) networks and the Media Engine, while Media Engine handles media streaming as well Traversal Using Relay Network Address Translation (TURN) functionality.

Although these two sub components work together as a single solution, and cannot be used independently, they employ completely different installation models. For instance, Signaling Engine is currently certified on a single operating system, while Media Engine supports a selection of bare metal servers. Likewise, Signaling Engine utilizes a graphical installer, while Media Engine is a command line driven installation. In addition, each component requires additional independent configuration and post installation steps. For those reasons, you must pay careful attention to the instructions contained in this guide.

Overview of the WebRTC Session Controller Installation Procedure

The installation procedure follows these steps:

1. Plan your installation. When planning your installation, you do the following:
   - Determine the scale of your implementation, for example, a small test system or a large production system.
   - Determine how many physical machines you need, and which software components to install on each machine.
   - Plan the system topology, for example, how the system components connect to each other over the network.

See "Planning Your WebRTC Session Controller Installation" for more information.

2. Review system requirements. System requirements include:

   - Hardware requirements, such as disk space
   - System software requirements, such as operating system (OS) versions and OS patch requirements, and Java Virtual Machine (JVM) process requirements (such as memory settings)
   - Information requirements, such as IP addresses and host names

   See "WebRTC Session Controller System Requirements" for more information.

3. Perform pre-installation tasks including assigning IP addresses to network assets and installing necessary support software.

   See "WebRTC Session Controller Pre-Installation Tasks" for more information.

4. Install WebRTC Session Controller Signaling Engine.

   See "Installing WebRTC Session Controller Signaling Engine" for more information.

5. Configure a WebRTC Session Controller Signaling Engine domain.

   See "Creating and Configuring a WebRTC Session Controller Signaling Engine Domain" for more information.

6. Perform Signaling Engine post-installation tasks.

   See "WebRTC Session Controller Signaling Engine Post-Installation Tasks" for more information.

7. Install WebRTC Session Controller Media Engine.

   See "Installing WebRTC Session Controller Media Engine" for more information.

8. Perform Media Engine post-installation tasks.

   See "WebRTC Session Controller Media Engine Post-Installation Tasks" for more information.

9. Troubleshoot any installation issues.

   See "Troubleshooting a WebRTC Session Controller Installation" for more information.

Ensuring a Successful Installation

The WebRTC Session Controller installation should be performed by qualified personnel. You must be familiar with both Signaling Engine and Media Engine software and the operating systems on which you are installing the software.

Follow these guidelines:

- As you install each component; for example, the JDK and WebRTC Session Controller Signaling Engine, verify that the component installed successfully before continuing the installation process.
- Pay close attention to the system requirements. Before you begin installing the software, make sure your system has the required base software. In addition, make sure that you know all of the required configuration values, such as host names and port numbers.

- As you create new configuration values, write them down. In some cases, you might need to re-enter configuration values later in the procedure.

Planning the Network and Hardware Setup

Before you can install WebRTC Session Controller, you must gather some information about your system and decide on the directories in which to install the software. You need to know:

- The network names or IP addresses of the machines on which you are going to install the following components:
  - The Signaling Engine
  - The Media Engine
- For machines hosting Signaling Engine installations, the directory on each machine which will serve as your *Middleware_home* directory. This directory serves as a repository for common files that are used by WebRTC Session Controller products installed on the same machine, such as WebLogic Server and a Java Development Kit.

  The files in the *Middleware_home* directory are essential to ensuring that software operates correctly on your system. They:
  - Facilitate checking of cross-product dependencies during installation
  - Facilitate Service Pack installation
- Passwords for administrative users on all server types.

WebRTC Session Controller has been tested to run on specific hardware and software platforms. "WebRTC Session Controller System Requirements" outlines supported configurations in detail. Unless your installation has been specified differently in cooperation with Oracle, only those configurations are supported.

Directory Placeholders Used in This Guide

Table 1–1 lists placeholders that are used in this guide to refer to the directories that contain WebRTC Session Controller system components.

Table 1–1 Directory Placeholders

Placeholder	Directory
Central_inventory_location	The directory in which the Oracle inventory file lives. The Oracle inventory lists all Oracle software installed on a machine.
Domain_home	The location of a configured WebRTC Session Controller WebLogic domain.
Middleware_home	The storage location for common WebRTC Session Controller files.
Oracle_home	The storage location for common Oracle software files.
temp_dir	A temporary directory into which you extract the installation images for Signaling and Media engine.
WSC_home	The directory in which WebRTC Session Controller is installed.

2 Planning Your WebRTC Session Controller Installation

This chapter provides information about planning your Oracle Communications WebRTC Session Controller installation.

About Planning Your WebRTC Session Controller Installation

When planning a WebRTC Session Controller installation, you consider how many physical servers can handle your subscriber base and how many signaling and media server nodes to include in your cluster.

About Development Systems, and Production Systems

The difference between a WebRTC Session Controller development system and a WebRTC Session Controller production system is only the number of machines in the system. You install the same components in a test system that you install in a production system.

See "WebRTC Session Controller System Requirements" for information about required hardware and software.

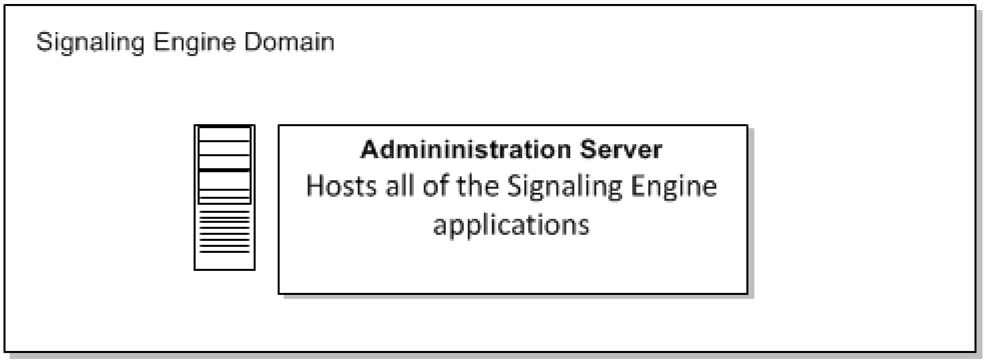
Planning Your Signaling Engine Installation

The following section describes recommended Signaling Engine installation topologies.

Understanding Signaling Engine Installation Topologies

Figure 2–1 shows simple Signaling Engine installation topology, a domain with a single Administration server. The Administration server hosts all of the Signaling Engine applications on a single machine.

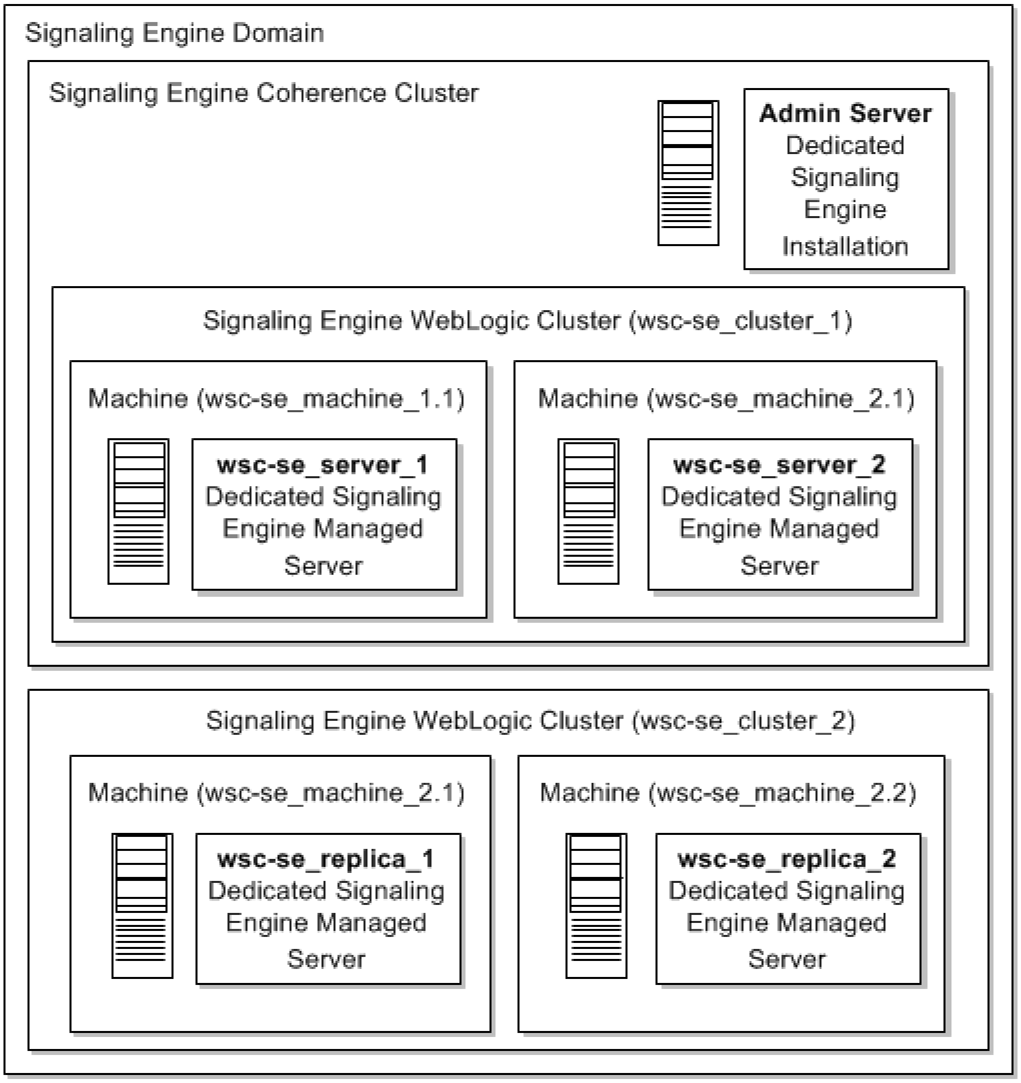
Figure 2–1 Basic Single Server Signaling Engine Domain



While a simple single host configuration is sufficient for development and Proof of Concept (PoC) installations, for production systems, a robust, fault-tolerant system is required.

Figure 2–2 shows a fault tolerant Signaling Engine installation topology. In this topology, the Administration server is installed on a separate machine from the two independently hosted clustered Managed Servers. In addition, there are two separately hosted Replicas. The Managed Servers can be separated geographically for additional fault tolerance, and additional replicas can be added as required.

Figure 2–2 Fault Tolerant Signaling Engine Domain


Signaling Engine Domain
Signaling Engine Coherence Cluster
Admin Server
Dedicated Signaling Engine Installation
Signaling Engine WebLogic Cluster (wsc-se_cluster_1)
Machine (wsc-se_machine_1.1)
wsc-se_server_1
Dedicated Signaling Engine Managed Server
Machine (wsc-se_machine_2.1)
wsc-se_server_2
Dedicated Signaling Engine Managed Server
Signaling Engine WebLogic Cluster (wsc-se_cluster_2)
Machine (wsc-se_machine_2.1)
wsc-se_replica_1
Dedicated Signaling Engine Managed Server
Machine (wsc-se_machine_2.2)
wsc-se_replica_2
Dedicated Signaling Engine Managed Server

Each element in this topology illustration is described in Table 2–1.

Table 2–1 Description of the Elements in the Signaling Engine Server and Coherence Standard Installation Topology

Element	Description and Links to Additional Documentation
Signaling Engine Domain	A logically related group of Java components (in this case, the Administration Server, Managed Signaling Engines, and other related software components). For more information, see "`Understanding Domains`" in *Understanding Oracle WebLogic Server*.
Administration Server	The central control entity of a domain which maintains the domain's configuration objects and distributes configuration changes to Managed Servers. For more information, see "`Administration Server`" in *Understanding Oracle WebLogic Server*.

Table 2–1 (Cont.) Description of the Elements in the Signaling Engine Server and Coherence Standard Installation Topology

Element	Description and Links to Additional Documentation
Cluster	A collection of multiple Signaling Engine instances running simultaneously and working together. For more information, see "Managed Servers and Managed Server Clusters" in *Understanding Oracle WebLogic Server*.
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are also the logical glue between Signaling Engine Managed Servers and the Node Manager; in order to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
Managed Server	Host for your applications, application components, Web services, and their associated resources. For more information, see "Managed Servers and Managed Server Clusters" in *Understanding Oracle WebLogic Server*.

WebRTC Session Controller Signaling Engine Coherence Planning

WebRTC Session Controller Signaling Engine nodes are based on Oracle Coherence. Decide how to configure Oracle Coherence settings for your WebRTC Session Controller Signaling Engine topology, for example, how many nodes to add to the cluster when a node failure occurs. For more information, see "Configuring and Managing Coherence Clusters" in *Administering Clusters for Oracle WebLogic Server*.

Planning Your Media Engine Installation

Media Engine nodes are installed on certified bare metal servers. Additional nodes can be added as required to support greater volumes of media traffic. For more information see *Oracle Communications Application Session Controller System Installation and Commissioning Guide*.

About Installing a Secure System

In a production system, you must ensure that communication between components and access to the system servers are secure. For information about choices for installing a secure system, see *Oracle Communications WebRTC Session Controller Security Guide*.

3 WebRTC Session Controller System Requirements

This chapter describes the software, hardware, and information requirements for Oracle Communications WebRTC Session Controller.

Software Requirements

This section describes the required software for the two WebRTC Session Controller sub components, Signaling Engine and Media Engine.

Signaling Engine Software Requirements

Signaling Engine is certified on Oracle Linux x64 version 6 or above running either natively or as a part of Oracle VM Server.

In addition, Signaling Engine requires a 64-bit Java Development Kit (JDK) version 1.7.0_25 or later.

Note: The following JDKs are not supported by Signaling Engine:

- Any OpenJDK
- Oracle JDK version 1.7.0_40

Media Engine Software Requirements

The Media Engine is a complete software stack, comprising a customized Linux kernel, infrastructure components, and the Media Engine application itself. Since Media Engine is self-contained, there are no additional software requirements.

About Critical Patch Updates

WebRTC Session Controller is supported on all Oracle Critical Patch Updates. You should install all Critical Patch Updates as soon as possible.

To download Critical Patch Updates, find out about security alerts, and enable email notifications about Critical Patch Updates, see the Security topic on Oracle Technology Network:

http://www.oracle.com/technetwork/topics/security/whatsnew/index.html

Hardware Requirements

This section describes the required hardware for the two WebRTC Session Controller sub components, Signaling Engine and Media Engine.

Signaling Engine Hardware Requirements

The number and configuration of the computers that you employ for your Signaling Engine installation depend on the scale and the kind of deployment you have planned according to your charging requirements. You will need to work with your performance team to determine your sizing requirements.

Signaling Engine has similar requirements to Oracle WebLogic Server 12c. The following items are required in addition to the basic WebLogic Server requirements:

- Gigabit Ethernet connections are required between engine and SIP data tier servers for all production deployments.
- Dual network interface cards (NICs) are required to provide fail-over capabilities in a production environment.
- Additional RAM is required to support the throughput requirements of most production installations.

Note: Each Transport Control Protocol (TCP) WebSocket connection requires approximately 14 kilobytes of RAM.

Media Engine Hardware Requirements

Media Engine is certified to run on the following server hardware:

- HP Proliant model DL360 Gen8
- Sun Netra X3-2

While Media Engine may run on other configurations, you are likely to run into disk controller as well as networking controller issues.

The number of physical or virtual servers will depend upon your particular environment load, but at a minimum each Media Engine server requires:

- Gigabit Ethernet connections
- 4 GB of RAM
- At least 50 GB of free hard disk space
- 64-bit Intel processor with two CPU cores

4

WebRTC Session Controller Pre-Installation Tasks

This chapter describes pre-installation tasks for Oracle Communications WebRTC Session Controller.

About Pre-Installation Tasks

You must perform certain tasks before installing WebRTC Session Controller. Pre-installation tasks are broken down into the following categories:

- General Pre-Installation Tasks: Tasks that are not specific to either the WebRTC Session Controller Media Engine (Media Engine) or WebRTC Session Controller Signaling Engine (Signaling Engine) components but that are required for a functioning installation.
- Signaling Engine Pre-Installation Tasks: Tasks that you must perform before you install Signaling Engine.

> **Note:** Since Media Engine is a complete self-contained software stack, there are no required pre-installation tasks.

General Pre-Installation Tasks

Before continuing, you must complete the following general pre-installation tasks:

- You should allocate IP addresses for Media Engine and Signaling Engine interfaces, including:
  - Public-facing external interfaces
  - Internal-facing interfaces
  - Intra-system interfaces for private signaling between Media Engine and Signaling engine nodes
  - If required by your organization, a separate systems management interface
- You should determine which logical interfaces map to which physical interfaces on each server.
- You should have access to a Domain Name Service (DNS) and Network Time Protocol (NTP) servers.
- Optionally you can assign fully qualified domain names to each server.
- You should make a note of any required static routes.

- You should download the WebRTC Session Controller software.

  To obtain the WebRTC Session Controller Signaling Engine Software:

  1. Download the WebRTC Session Controller software from the Oracle software delivery Web site, located at:

     `http://edelivery.oracle.com`

     and save it to a temporary directory (*temp_dir*).

  2. Unzip the WebRTC Session Controller installation files.

Signaling Engine Pre-Installation Tasks

Before installing Signaling Engine software, you must complete the following pre-installation tasks:

- Install a Java Development Kit
- Create the Signaling Engine User Account
- Choose an Installation Directory

Install a Java Development Kit

Install an Oracle Java Development Kit (JDK) 1.7.0_25 or greater and add it to your PATH environment variable. The JRE is required for the installer process.

You can download a JDK from the Java SE Development Kit 7 Downloads page: `http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html`.

See "`JDK 7 Installation for Linux Platforms`" in the Oracle Java SE Documentation for instructions on installing a JDK.

> **Note:** Signaling Engine is not compatible with the OpenJDK installed by default in development installations of Oracle Linux 6, nor is it compatible with Oracle JDK 1.7.0_40.

Create the Signaling Engine User Account

Create the user account that is to be the primary user running Signaling Engine in your environment.

It is required that all machines have the same user name configured.

Note the user name; you will be required to specify the user name during the Signaling Engine installation process.

To create the user account:

1. Log in to the driver machine.
2. Enter the following command:

   ```
   useradd passwd user_name
   ```

   where *user_name* is the name of the user and *passwd* is the password for the user.

See the discussion in your Linux documentation for more information about the **useradd** command.

Choose an Installation Directory

When you install WebRTC Session Controller, you are prompted to specify a Middleware home directory. This directory serves as a repository for common files that are used by multiple Fusion Middleware products installed on the same machine. For this reason, the Middleware home directory can be considered a central support directory for all the Fusion Middleware products installed on your system.

The files in the Middleware home directory are essential to ensuring that WebRTC Session Controller and WebLogic Server operate correctly on your system. They facilitate checking of cross-product dependencies during installation.

For more information on choosing an installation directory, see "`Understanding the Oracle WebLogic Server and Coherence Directory Structure`" in *Installing and Configuring Oracle WebLogic Server and Coherence*.

Next Steps

After you have completed pre-installation tasks you can install Signaling Engine. See "Installing WebRTC Session Controller Signaling Engine" for instructions.

5

Installing WebRTC Session Controller Signaling Engine

This chapter describes how to install Oracle Communications WebRTC Session Controller Signaling Engine (Signaling Engine).

Before you install Signaling Engine, you must complete all pre-installation tasks described in "WebRTC Session Controller Pre-Installation Tasks".

About the GUI Installation and Silent Installation

You can install Signaling Engine by using the GUI installation or the silent installation. The silent installation procedure enables you to perform a non-interactive installation of Signaling Engine. You can use the silent installation to install Signaling Engine quickly on multiple systems.

The silent installer uses a response file in which you specify installation settings. To obtain the response file, you first run the GUI installation, and choose to save a response file.

For installation instructions, see the following sections:

- Installing Signaling Engine Using the GUI Installation
- Installing Signaling Engine Using the Silent Installation

Installing Signaling Engine Using the GUI Installation

To install Signaling Engine:

1. If you have not done so already, download and unzip the WebRTC Session Controller software. See "General Pre-Installation Tasks" for instructions.
2. Log in to the system on which you want to install Signaling Engine.
3. The installer requires that a certified JDK already exists on your system. For more information, see "Install a Java Development Kit".
4. Go to the directory where you downloaded the installation program.
5. Launch the installation program by invoking `java -jar` from the JDK directory on your system, as shown in the example below:

   ```
   java -jar wsc_generic.jar
   ```

   If no other Oracle products are installed on the system, the Installation Inventory screen appears. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location.

For more information about the central inventory, see "Understanding the Oracle Central Inventory" in *Installing Software with the Oracle Universal Installer*.

If an Installation Inventory already exists, the WebRTC Session Controller Installation Welcome window appears.

6. Click **Next**.

   The WebRTC Session Controller Oracle Installation Location window appears.

   Use this screen to specify the location of your Oracle home directory.

   For more information about Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware*.

7. Click **Next**.

   The Installation Type window appears.

   There is only one installation type for WebRTC Session Controller, "**WebRTC Session Controller Installation**," and it is selected by default.

8. Click **Next**.

   The Prerequisite Checks window appears.

   This screen verifies that your system meets the minimum necessary requirements.

   If there are any warnings or errors, make sure that your environment meets all the necessary prerequisites. See "WebRTC Session Controller System Requirements" for more information.

9. Click **Next**.

   The Security Updates window appears.

   If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates.

   If you do not have one and are sure you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.

10. Click **Next**.

    The Installation Options screen appears.

    Use this screen to verify the installation options you selected. If you want to save these options to a response file, click **Save Response File** and provide a location and name for the response file. Response files can be used later in a silent installation situation.

    Note: If you only want to save a response file, you can exit the installation at this time.

    For more information about silent mode installation, see "Installing Signaling Engine Using the Silent Installation".

11. Review the selections you have made, and click **Install**.

    The Installation Progress window appears and the installation begins.

Note: After the installation begins, if you click **Cancel**, the installation process stops but the files that are already copied are not removed.

---

12. When the installation is completed, click **Next**.

    The Installation Complete window appears.

    The next step is to launch the configuration wizard to create your WebLogic domain. There are two ways to do this:

    - Select **Automatically Launch the Configuration Wizard** on this screen. After you click **Finish** to close the installer, the configuration wizard is started and you can begin to configure your domain. If you choose to do this, proceed to "Configuring a WebRTC Session Controller Domain".
    - Do not select **Automatically Launch the Configuration Wizard** on this screen. After you click **Finish** to close the installer, you must manually start the configuration wizard to begin configuring your domain. If you choose to do this, proceed to "Starting the Configuration Wizard".

Installing Signaling Engine Using the Silent Installation

The silent installation uses a response file in which you have set installation information. To obtain the response file, you run the GUI installer up until the Installation Options pane appears. You can then save a response file that contains the key and values pairs based on the values that you specify during the GUI installation. You can then copy and edit the response file to create additional response files for installing Signaling Engine on additional machines.

Creating a Response File

The response file must contain the key and value pairs for the mandatory parameters the installer must use for the Signaling software component you install. All information prompted in the GUI installation is associated with mandatory parameters.

To create a response file:

1. Run the GUI installation for Signaling Engine. See "Installing Signaling Engine Using the GUI Installation" for instructions.
2. When the Installation Options screen appears, click **Save Response File** and provide the location and name for the response file.
3. Modify the response file you copied by specifying the key and value information for the parameters you want in your installation.
4. Save and close the response file.

Note: For information on response file key and value pairs, see "`Using the Oracle Universal Installer in Silent Mode`" in *Installing Software with the Oracle Universal Installer*.

Performing a Silent Installation

To perform a silent installation:

1. Create a response file. See "Creating a Response File" for instructions.
2. Download and unzip the WebRTC Session Controller software on the machine on which you will run the silent installation. See "General Pre-Installation Tasks" for instructions.
3. Copy the response file you created to the machine on which you will run the silent installation.
4. On the machine on which you will run the silent installation, run the following command:

   ```
   java -jar wsc_generic.jar -silent -responseFile fullpathtoresponsefile
   ```

   Where *fullpathtoresponsefile* is the full path including the filename of your response file

   For example:

   ```
   java -jar wsc_generic.jar -silent -responseFile /home/user/responsefile.txt
   ```

The WebRTC Session Controller Installer checks for all required software and writes errors to a log file if it detects any missing or unavailable components, or if there are any connectivity-related issues.

See "Troubleshooting a Signaling Engine Installation" for information about WebRTC Session Controller Signaling Engine installer logs.

Next Steps

After you install Signaling Engine, you must configure a Signaling Engine domain. See "Creating and Configuring a WebRTC Session Controller Signaling Engine Domain" for instructions.

> **Note:** To uninstall Signaling Engine, you run the command *Oracle_home***/oui/bin/desinstall.sh** and click **Next** and then click **Deinstall**. See "`Oracle Universal Installer Deinstallation Screens`" in *Oracle Fusion Middleware Installing with the Oracle Universal Installer* for more information.
>
> You will have to remove the *Oracle_home* directory before you can reinstall the software.

6 Creating and Configuring a WebRTC Session Controller Signaling Engine Domain

This chapter describes the steps required to create your WebRTC Session Controller Signaling Engine (Signaling Engine) domain after your software has been successfully installed.

About Domains and Domain Configuration

After you install the Signaling Engine software, you must create a domain for your deployment. Before continuing in this chapter, you need to understand WebLogic domains and clustering, and the domain topologies available for use with WebRTC Session Controller Signaling Engine

- To learn about WebLogic domains and clustering, see "`WebLogic Server Domains`" and "`WebLogic Server Clustering`" in Understanding Oracle WebLogic Server.
- To learn about the domain topologies available for use with WebRTC Session Controller Signaling Engine, see "Understanding Signaling Engine Installation Topologies".

Configuring Your Signaling Engine Domain

This section provides instructions for creating a WebRTC Session Controller domain using the configuration wizard. For more information on other methods available for domain creation, see "`Additional Tools for Creating, Extending, and Managing WebLogic Domains`" in *Creating Domains Using the Configuration Wizard*.

The following topics are covered in this section:

- About Signaling Engine Domain Types
- Starting the Configuration Wizard
- Configuring a WebRTC Session Controller Domain

About Signaling Engine Domain Types

There are a selection of domain templates to choose from but only two are relevant to Signaling Engine:

- **Oracle Communications WebRTC Session Controller Replicated Domain**

  The Replicated Domain template enables you to create a replicated WebRTC Session Controller Signaling Engine domain. The Replicated Domain topology is designed for use with WebRTC applications that require high levels of scalability, availability, and performance.

- **Oracle Communications WebRTC Session Controller Basic Domain**

  The Basic Domain template enables you to create a simple Signaling Engine domain. Such a domain configuration can be used during development where it is more convenient to deploy and test applications on a single server.

In addition, each domain type can be extended to add Diameter support. For more information, see the discussion of WebRTC Session Controller Diameter Rx to Policy Charging and Rules (PCRF) configuration in the *Oracle Communications WebRTC Session Controller System Administrator's Guide*.

Starting the Configuration Wizard

To begin domain configuration, navigate to the *Oracle_home*/**oracle_common/common/bin** directory and start the Fusion Middleware Configuration Wizard:

```
./config.sh
```

> **Note:** If, while installing Signaling Engine using the GUI installation wizard, you checked the Automatically Launch the Configuration Wizard check box, the Domain Configuration wizard will already be running.

Configuring a WebRTC Session Controller Domain

Follow the instructions in this section to configure the domain using the Configuration Wizard.

1. On the Configuration Type screen, select **Create a New Domain**.

   In the Domain Location field, specify your Domain home directory.

   It is recommended that you locate your Domain outside the Oracle home directory. This directory structure will help you avoid issues when you need to upgrade or re-install your software.

   > **Tip:** More information about the Domain home directory can be found in "Choosing a Domain Home" in *Planning an Installation of Oracle Fusion Middleware*.
   >
   > More information about the other options on this screen can be found in "Configuration Type" in *Creating Domains Using the Configuration Wizard*.

2. Click **Next**.

   The Templates window appears.

3. On the Templates screen select one of the following templates:

   - **Oracle Communications WebRTC Session Controller Replicated Domain**

     Selecting this template also selects **WebLogic Coherence Cluster Extension**.

   - **Oracle Communications WebRTC Session Controller Basic Domain**

     Selecting this template also selects **Basic WebLogic SIP Server Domain** and **WebLogic Coherence Cluster Extension**.

Note: The template **Basic WebLogic Server Domain** is selected by default and cannot be deselected.

More information about the options on this screen can be found in "`Templates`" in *Creating Domains Using the Configuration Wizard*.

4. Click **Next**.

   The Administrator Account screen appears.

   On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain. This account is used to connect to the domain's Administration Server.

   Tip: You must make a note of the user name and password you choose to enter here; you will need this in order to be able to start and access the Administration Server.

5. Click **Next**.

   The Domain Mode and JDK screen appears.

   On the Domain Mode and JDK screen:

   - Select **Development** or **Production** in the Domain Mode field.
   - Select **Oracle Hotspot JDK** in the JDK field or choose a different supported JDK.

     See "Signaling Engine Software Requirements" for information on supported JDKs.

   Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

   Tip: In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. For more information, see "`Creating a Boot Identity File for an Administration Server`" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

6. Click **Next**.

   There are several advanced options you can choose to configure on the Advanced Configuration screen:

   - **Administration Server**

     Checking this option lets you configure the listen address of the Administration Server.

   - **Node Manager**

     Checking this option lets you configure Node Manager.

   - **Managed Servers, Clusters and Coherence**

     Checking this option lets you configure the Managed Servers, Clusters, and also lets you configure the machine and assign Managed Servers to the machine.

Tip: If you want to configure dynamic clusters, see the following:

- "Overview of Dynamic Clusters" in *Understanding Oracle WebLogic Server*.
- "Creating Dynamic Clusters" in *Administering Clusters for Oracle WebLogic Server*.

- **Deployments and Services**

  Checking this option lets you customize how application deployments and services are targeted to servers and clusters.

Check the advanced options you wish to configure.

Note: If you are configuring a Oracle Communications WebRTC Session Controller Replicated Domain in **Production Mode**, you must select the following advanced options:

- **Administration Server**
- **Managed Servers, Clusters and Coherence**

Failure to configure those options will cause the WebLogic server to fail to start.

7. If you have not checked any advance options, click **Next** and continue to step 8. Otherwise, click **Next** and follow this sub procedure:

   a. If you have chosen the Administration Server advanced option, the Administration Server screen appears.

   For more information on the options available on this screen, click the **Help** button and refer to the Wizard's online help.

   Note: If you are configuring a **Oracle Communications WebRTC Session Controller Replicated Domain** in **Production Mode**, you must enter the **Listen Address** for the Administration Server.

   Make any updates required and click **Next**.

   b. If you have chosen the Node Manager advanced option, the Node Manager screen appears.

   For more information on the options available on this screen, click the **Help** button and refer to the Wizard's online help.

   Make any updates required and click **Next**.

   c. If you have chosen the Managed Servers, Clusters and Coherence advanced option, the following four screens will appear in succession: Managed Servers, Clusters, Coherence Clusters, and Machines.

   For more information on the options available on these screens, click the **Help** button and refer to each screen's online help.

Note: If you are configuring a **Oracle Communications WebRTC Session Controller Replicated Domain** in **Production Mode**, you must select or enter the **Listen Address** for each engine and replica in the Managed Servers screen.

You must also enter the **Cluster Address** for the engine cluster in the Clusters screen.

Make any updates required and click **Next** on each screen.

d. If you have chosen the Deployments Targeting advanced option, the Deployments Targeting screen appears.

For more information on the options available on this screen, click the **Help** button and refer to the Wizard's online help.

Make any updates required and click **Next**.

8. The Configuration Summary screen appears.

   The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.

   You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

   Domain creation will not begin until you click **Create**.

9. The Configuration Success screen will show the following items about the domain you just configured:

   - Domain Location
   - Administration Server URL

   You must make a note of both items as you will need them to start the servers and access the Administration Server.

   Click **Finish** to close the configuration wizard.

Starting the Signaling Engine Servers

After configuration is complete, in order to access the tools with which you can manage your domain, you must start the necessary servers. See the following topics for more information:

- Starting the Node Manager
- Starting the Administration Server
- Starting the Managed Servers

Starting the Node Manager

To start your per-domain Node Manager, go to the `*Domain_home*/bin` directory.

Start Node Manager as shown below, using `nohup` and `nm.out` as an example output file:

```
nohup ./startNodeManager.sh > nm.out&
```

Note: It is recommended that you install Node Manager to run as a startup service. This allows Node Manager to start up automatically each time the system is restarted.

For more information, see "Running Node Manager as a Startup Service" in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Administration Server

To start the Administration Server, go the *Domain_home*/bin directory and run:

```
./startWebLogic.sh
```

If you selected **Production Mode** on the Domain Mode and JDK screen in step 5, you will be prompted for the login credentials of the Administrator user as provided on the Administrator Account screen in step 4.

Tip: For more information about starting the Administration Server, see "Starting and Stopping Administration Servers" in *Administering Oracle Fusion Middleware*.

In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. For more information, see "Creating a Boot Identity File for an Administration Server" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

Starting the Managed Servers

To start the Managed Servers, go the *Domain_home*/bin directory and run the following command:

```
./startManagedWebLogic.sh managed_server_name admin_server_url
```

Replace *managed_server_name* with the name of the Managed Server you want to start.

Replace *admin_server_url* with the full URL of the Administration Server, as provided on the Configuration Success screen in step 9.

Below are sample commands used to start wls_server_1 and wls_server_2 on UNIX operating systems:

```
./startManagedWebLogic.sh wsc-se_server_1 t3:\\host.example.com:7001 &
./startManagedWebLogic.sh wsc-se_server_2 t3:\\host.example.com:7001 &
```

Tip: For more information about starting Managed Servers, see "Starting and Stopping Managed Servers" in *Administering Oracle Fusion Middleware*.

Next Steps

After you have configured the Signaling Engine domain, you must complete Signaling Engine post-installation tasks. See "WebRTC Session Controller Signaling Engine Post-Installation Tasks" for instructions.

7

WebRTC Session Controller Signaling Engine Post-Installation Tasks

This chapter provides instructions for Oracle Communications WebRTC Session Controller Signaling Engine (Signaling Engine) post-installation tasks.

Before continuing, you must complete the tasks in the following chapters:

- Installing WebRTC Session Controller Signaling Engine
- Creating and Configuring a WebRTC Session Controller Signaling Engine Domain

Overview of Signaling Engine Post-Installation Tasks

After installing Signaling Engine, you must complete the following post-installation tasks:

- Enable DNS Server Lookup
- Set the SIP Proxy Server and Registrar IP Address
- Configure WebRTC Session Controller Authentication
- Configure the Coherence Security Framework

Before continuing, you must start your Signaling Engine servers. See "Starting the Signaling Engine Servers" for more information.

Enable DNS Server Lookup

You must enable DNS server lookup on your Signaling Engine installation to ensure that the **maddr** parameter in SIP headers is processed properly.

To enable DNS server lookup:

1. Start your Signaling Engine servers if they are not already running. See "Starting the Signaling Engine Servers" for more information.
2. Navigate to the WebLogic Server administration console and log in with your administrator username and password:

   ```
   http://hostname:port/console
   ```

   > **Note:** The default administration console port is 7001.

3. In the Domain Structure pane, select **SipServer**.

4. Select the Configuration tab and then select the General sub tab.
5. In the General sub tab, scroll down and select the **Enable DNS Server Lookup** option.
6. Click **Save**.
7. Log out of the administration interface.

Set the SIP Proxy Server and Registrar IP Address

To set the SIP proxy server and registrar IP address:

1. Start your Signaling Engine servers if they are not already running. See "Starting the Signaling Engine Servers" for more information.
2. Navigate to the WebRTC Session Controller Signaling Engine console and log in with your administrator username and password:

   ```
   http://hostname:port/wsc-console
   ```

   > **Note:** The default Signaling Engine console port is 7001.

3. Select the **Script Library** tab.
4. Click **Lock and Edit**.
5. In the Script Library pane, find the following lines and replace *hostname* and *port* with the hostname and port of your SIP proxy server and registrar:

   ```
   /** Proxy/Registrar URI. */
   public static final PROXY_SIP_URI = "sip:hostname:port"
   ```

6. Click **Validate Library** to ensure you introduced no errors. Fix any errors that are reported.
7. Click **Commit** to save your changes.

Configure WebRTC Session Controller Authentication

You must configure WebRTC Session Controller to provide authentication for WebRTC clients, even if you only want to allow unauthenticated access. For more information, see the discussion on configuring WebRTC Session Controller authentication in *Oracle Communications WebRTC Session Controller System Administrator's Guide*.

Configure the Coherence Security Framework

If you have created a clustered domain, you must enable the Coherence Security Framework. For instructions, see "Enabling the Oracle Coherence Security Framework" in *Securing Oracle Coherence*.

Next Steps

You must install WebRTC Session Controller Media Engine servers. See "Installing WebRTC Session Controller Media Engine" for instructions.

8

Installing WebRTC Session Controller Media Engine

This chapter describes how to install Oracle Communications WebRTC Session Controller Media Engine (Media Engine).

Before you install Media Engine, you must complete the tasks in the following chapters:

- Installing WebRTC Session Controller Signaling Engine
- Creating and Configuring a WebRTC Session Controller Signaling Engine Domain
- WebRTC Session Controller Signaling Engine Post-Installation Tasks

About Installing Media Engine

Table 8–1 describes the Media Engine software distribution formats available from https://edelivery.oracle.com. Download the one that is appropriate for your environment.

Table 8–1 Media Engine Software Distribution Formats

Name	Description
Oracle Communications WebRTC Session Controller 7.0 - ME Software USB Image	A Media Engine distribution suitable for use with the BMC tool to create a USB software installation stick. Only download this software archive if you are installing Media Engine using a USB stick. The BMC tool is included in the USB image archive. For more information on the BMC tool, see the discussion on creating and commissioning USB sticks in the *Oracle Communications Application Session Controller Installation and Commissioning Guide*.
Oracle Communications WebRTC Session Controller 7.0 - ME Software Supertar	A Media Engine installation file used for upgrading an existing installation.
Oracle Communications WebRTC Session Controller 7.0 - ME Software ISO Image	A Media Engine distribution suitable for creating a bootable CD/DVD image. Only download this software archive if you are installing Media Engine using a CD or DVD drive.

Table 8–1 (Cont.) Media Engine Software Distribution Formats

Name	Description
Oracle Communications WebRTC Session Controller 7.0 - ME Software Xen Server Image	A Media Engine distribution suitable for loading on a Xen virtualization server. Only download this software archive if you are installing Media Engine on a Xen hypervisor.

For more information on Media Engine installation scenarios, see the *Oracle Communications Application Session Controller System Installation and Commissioning Guide*.

For details on supported bare metal servers as well as general hardware requirements, see "Media Engine Hardware Requirements".

Media Engine Installation and Network Configuration

Once you have created the Media Engine installation image, you first attach the bootable image to your hardware, run the Media Engine system installation, and configure basic networking information.

To install Media Engine and configure networking information:

1. Insert the bootable CD/DVD or attach the bootable USB drive to your server.
2. Power on the machine.
3. Once the installation image has booted, you are prompted to reformat the primary hard disk. Select **yes**.

   When the installation is complete, the machine will reboot.
4. At the login prompt, enter **root** and press Enter.

   The password prompt appears.

   > **Note:** While not required for the initial login, the root password should be changed as soon as possible after installation. For more information, see the discussion about configuring permissions, users and authorization in *Oracle Communications Application Session Controller System Administration Guide*.

5. Press Enter.

   The application-level user name prompt appears.
6. Press Enter. No application-level user name is required upon initial login.

   The application-level password prompt appears.
7. Press Enter.

   After you have logged in, the Media Engine automatically reboots. Once the system has finished rebooting, you must run the setup script to create a management IP interface.
8. At the console prompt, type `config setup` to access the basic configuration objects and properties you must modify on your first network interface to ping the Media Engine and access it via SSH on port 22, and HTTPS on port 443.

> **Note:** Once you have configured basic access to your first network interface, you can access the system and use the configuration commands to complete your system configuration. For more information on configuration objects and properties, see the *Oracle Communications OS-E Objects and Properties Reference Guide*.

9. When prompted for a **hostname**, enter the name you want to use for the host and press Enter.
10. When prompted for an **interface**, enter the name of the ethernet interface, for example, eth1, and press Enter.
11. When prompted for an **ip-address**, enter the IP address and subnet mask of the host, for example, 192.168.12.123/24, and press Enter.
12. When asked if you want to enable **ssh** access to the Media Engine host, enter `y` and press Enter.
13. When asked if you want to enable **web** access to the Media Engine host, enter `y` and press Enter.
14. When prompted for a **route**, enter `default` for the ethernet interface route and press Enter.
15. When prompted for a default **gateway**, enter the IP address of the host's default gateway and press Enter.
16. When prompted for the text you want to use for the host's **prompt**, enter a string of your choice and press Enter.
17. When prompted for **media-sessions**, enter the session capacity per feature to match what you have purchased, and press Enter.
18. When you are asked to commit this setup script, enter y.
19. When you are asked to update the startup configuration, enter y.

Editing and Installing a Media Engine Configuration File

You must edit the sample configuration file provided with Media Engine with details specific to your installation. See "Sample Media Engine Configuration File" for more information.

Once you have made the necessary changes to the configuration file, you must upload it to the Media Engine **/cxc** folder.

To upload the sample configuration file to the Media Engine **/cxc** folder:

1. Navigate to the Media Engine login page:

   `https://`*`hostname`*

   The login page appears.

2. Click **Login**.

   The Media Engine home page appears.

Note: No credentials are required to login until users, permissions, and authorization have been configured. For more information on configuring those objects and properties, see *Oracle Communications OS-E System Administration Guide*.

3. Select the **Tools** tab.
4. Click **Upload File**.
5. Click **Browse** and navigate to the sample configuration file.
6. Click **Update**.
7. Sign out of the Media Engine web page and open a SSH session with the Media Engine host.
8. Sign in to the Media Engine console.
9. Enter the following command to validate the configuration file:

   ```
   config validate /cxc/filename
   ```

10. Enter the following command to activate the configuration file:

    ```
    config replace /cxc/filename
    ```

11. Enter the following command to save the sample configuration file:

    ```
    config save /cxc/filename
    ```

Sample Media Engine Configuration File

Example 8–1 shows an annotated Media Engine configuration file. Variables that must be updated in this configuration file are indicated by the text, **SET THIS**.

Note: This is only an example. The exact contents of this configuration file can vary significantly depending upon your particular networking environment and requirements.

For more information on the objects and properties referenced in this example, see the *Oracle Communications OS-E Objects and Properties Reference Guide*.

Example 8–1 Annotated Media Engine Configuration File

```
#
# This is an example of a Media Engine cluster configuration with recommended configuration
# settings. Details will be different for each customer site and may differ from this template.
#
# Items marked with ----> SET THIS must be supplied
#

config cluster

#
# Standard box-level settings.
#
 config box 1
  set hostname SET THIS to the host name
```

```
  set timezone **SET THIS** to the host's timezone
  set name **SET THIS** to the box name
  set identifier **SET THIS** to the box's eth0 MAC address

#
# Interface backing the VRRP vx interface - see VRRP configuration below.
#
  config interface eth1
  return

#
# Heartbeat interface for other boxes in the cluster. The eth0 interface is recommended.
#
  config interface eth0
   config ip heartbeat1
    set ip-address static **SET THIS** to the IP address and network mask, for example, 192.168.1.1/24

#
# Web services are configured here so the load factor application can retrieve statistics
# for all boxes in the cluster.
#
    config web-service
    return
#
# Set up a Network Time Protocol (NTP) server. It is important that cluster time is correct, and
# this NTP server acts as backup in case an external NTP server is not available.
#
    config ntp-server
    return

    config vrrp-advertisements
    return
    config messaging
    return
   return
  return
  config cli
   set prompt **SET THIS** to a useful descriptor such as the box name
   set display scrolled
  return

#
# Set to an external NTP server; it is important that the cluster have the correct time.
#
  config ntp-client
   set server **SET THIS** to the external NTP server's IP, for example, 10.148.104.1
  return

#
# Increases the size of the kernel Address Resolution Protocol (ARP) cache.
#
  config os
   set arp-thresholds 4096 4096 8192
  return

#
# Sets the number of active ports available for media anchoring.
#
  config media-anchor-limits
   config port-limit CXC CXC
```

```
    set full-limit 800000
   return
  return
 return

#
# Standard box-level settings for an additional box.
#
 config box 2
  set hostname SET THIS to the box’s host name.
  set timezone SET THIS to box’s timezone.
  set name SET THIS to the box’s name.
  set identifier SET THIS to the box’s MAC address.

#
# Interface backing the Virtual Router Redundancy Protocol (VRRP) vx interface - see VRRP
# configuration below.
#
  config interface eth0
  return

#
# Heartbeat interface to the other box in the cluster
#
  config interface eth1
   config ip heartbeat2
    set ip-address static SET THIS to the local interface/netmask, for example, 192.168.1.2/24

#
# Web services are configured here so the load factor application can get access
# to statistics on every box in the cluster.
#
    config web-service
    return

    config vrrp-advertisements
    return
    config messaging
    return
   return
  return
  config cli
   set prompt SET THIS to the box name.
   set display scrolled
  return

#
# Set to an external NTP server; it’s important that the cluster have the correct time
#
  config ntp-client
   set server SET THIS to the external NTP server IP, for example, 10.148.104.1
   set server SET THIS to the internal NTP server on ip Heartbeat1, for example, 192.168.1.1
  return

#
# Increases the size of the kernel ARP cache.
#
  config os
   set arp-thresholds 4096 4096 8192
  return
```

```
#
# Sets the number of active ports available for media anchoring.
#
  config media-anchor-limits
   config port-limit CXC CXC
    set full-limit 800000
   return
  return
 return

#
# Cluster support
#
 set share-signaling-entries true
 set mirror-media-streams true

#
# Cluster interfaces using VRRP
#
 config vrrp

#
# management, web service and SIP interface
#
  config vinterface vx36

#
# Define a group of box interfaces tied to this cluster interface.
#
   set group 1
   set host-interface cluster\box 1\interface eth0
   set host-interface cluster\box 2\interface eth0

#
# We're doing "one-arm" routing, in that there are no separate public, private
# and management interfaces.
#
   config ip Service
    set ip-address static SET THIS to the IP address/netmask, for example, 10.1.1.1/16

#
# Management protocols.
#
    config ssh
    return
    config web
    return

#
# Required for the media control API between Signaling Engine and Media Engine.
#
    config web-service
    set min-spare-threads 4
    return

#
# SIP and media.  Note that Network Address Translation (NAT) is enabled.
#
    config sip
```

```
     set udp-port 5060 "" "" any 0
     set tcp-port 5060 "" "" any 0
     set tls-port 5061 "" "" any 0
     config ws-port 9080
     return
    return
    config icmp
    return
    config media-ports
    return

#
# Routing - default route for the box.
#
    config routing
     config route default
      set gateway SET THIS to the gateway IP address, for example, 10.1.0.1
     return
    return

#
# These are the default ports for the STUN server.  'disabled' refers to the TURN relay.
#
    config stun-server
     set port UDP 3478 disabled
    return
   return

#
# Configuration for the load factor application which supplies load factor
# values for the Media Engine cluster to the Signaling Engines.
#

   config ip loadfactor
    set ip-address static SET THIS to an IP address/netmask, for example, 10.1.1.2/16
    config web-service
     config virtual-host SET THIS to same static address for this ip, for example, 10.1.1.2
      config web-app-config /loadfactor
       set context-parameter ascMgmtHost SET THIS to the web services IP address on the
                                                  IP Service, for example, 10.1.1.1
       set context-parameter ascUsername SET THIS to the Media Engine user name in ME access
                                                  configuration with web services privileges
       set context-parameter ascPassword SET THIS to the password of the Media Engine user name
                                                  specified above
      return
     return
    return
   return
  return

#
# Sets the Media Engine to initiate a VRRP failover to backup boxes if the local Session
# Initialization Protocol (SIP) process goes dead
#
  set sip-dead-groups 1
 return
return

#
# Standard logging, and other options.
```

```
#
config services
 config event-log
  config file krnlsys.log
   set filter krnlsys debug
  return
  config file management.log
   set filter management info
  return
  config file access.log
   set filter access info
  return
  config file system.log
   set filter general info
   set filter sipSvr info
   set filter system info
   config advanced-filter
    set blocked-event irqbalance
   return
  return
  config file error.log
   set filter all error
  return
  config file db.log
   set filter db debug
  return
 return
 config data-locations
 return
 config storage-device
  set fail-threshold 200
 return

#
# Configure the collect command group specific to WebRTC for debugging.
#
 config collect
  config collect-group wsc-me
   set status-class version
   set status-class ice-state-status
   set status-class ice-dtls-status
   set status-class media-stream-stats
   set status-class media-stream-addresses
   set status-class kernel-rule-stats
   set status-class kernel-rule
   set description "WSC ME Debug"
  return
 return
return

#
# Enables and sets the master box for master services. Each master services can run on
# one box in the cluster.
#
config master-services

#
# Box for controlling the cluster configuration.
#
 config cluster-master
```

```
  set host-box cluster\box 1
  set host-box cluster\box 2
  set group 1
 return

#
# The internal cluster database for call-log debugging.
#
 config database
  set host-box cluster\box 2
  set host-box cluster\box 1
  set group 2
  set media enabled
 return

#
# Failover for media processing.
#
 config call-failover
  set host-box cluster\box 1
  set host-box cluster\box 2
  set group 3
 return

#
# Event service settings.
#
 config events
  set host-box cluster\box 1
  set host-box cluster\box 2
  set group 4
 return
return

config vsp
 set admin enabled

#
# Default session config; this is overridden/supplemented by policy rules and named
# session configs.
#
 config default-session-config

#
# By default anchor media and terminate calls where there has been no media activity
# for the specified period.
#
  config media
   set anchor enabled
   config nat-traversal
    set symmetricRTP true
   return
   set inactivity-timeout enabled "0 days 00:01:00"
   set rtp-stats enabled
  return

#
# We'll allow endpoints to use media encryption if they want.  reuse-key ensures
# that the key is not renegotiated when a re-INVITE is received.  Chrome uses SHA1-80,
# so we'll disabled SHA1-32.
```

```
#
 config in-encryption
   set mode allow
   set type multiple
   set priority-AES-128-CM-HMAC-SHA1-32 0
   set priority-AES-128-CM-HMAC-SHA1-80 1
   set reuse-key enabled
   set encryption-preferences 1 DTLS
   set encryption-preferences 2 RFC_3711
  return
  config out-encryption
   set mode allow
   set type multiple
   set reuse-key enabled
   set encryption-preferences 1 DTLS
   set encryption-preferences 2 RFC_3711
  return

#
# The only video codec supported by Chrome is VP8, so we'll get rid of anything else.
#
  config media-type
   set allowed-media-types video VP8
   set blocked-media-types video any
  return

#
# Minimizes the risk of Chrome and other endpoints choking on SDP attributes they don't understand.
# Pass SDP attributes unchanged.
#
  config sdp-regeneration
   set origin pass
   set add-rtpmaps enabled
   set pass-attribute candidate
   set pass-attribute ice-pwd
   set pass-attribute ice-ufrag
   set pass-attribute rtcp
   set pass-attribute rtcp-mux
   set pass-attribute ssrc
   set pass-attribute remote-candidates
   set pass-attribute fingerprint
   set pass-attribute rtcp-fb
   set pass-attribute wsc-session-id
  return

#
# We'll let calls in by default.
#
  config sip-directive
   set directive allow
  return

#
# Enables call log for debugging purposes.
#
  config log-alert
  return

#
# Makes sure we are using the RTP profile required by WebRTC.
```

```
#
  config header-settings
   config altered-body 1
    set altered-body "(.*)RTP/SAVP (.*)" "\1RTP/SAVPF \2" custom
   return
  return

#
# 3PCC is not just third party call control, it's the whole SIP state machine.
# Note that setting always-apply-req-uri-spec to disabled means we do the right
# thing when the request URI is changed during a call.
#
  config third-party-call-control
   set admin enabled
   set handle-refer-locally disabled
   set always-apply-req-uri-spec disabled
   set terminate-reinvite-locally enabled
  return

#
# The event-settings are required for the media control API.
#
  config event-settings
   set channel /wsc/agent/%$call.request-id%
   set named-variable-entry to
   set include-media-content enabled
  return

#
# In some cases, SSRC in SDP can cause trouble, so we'll strip it out,
# and add it in elsewhere if we need it.
#
  config in-sdp-attribute-settings
   set ssrc-in-sdp strip
  return
  config out-sdp-attribute-settings
   set ssrc-in-sdp strip
  return
 return

#
# DTLS settings.
#
 config tls
  config default-dtls-settings
   set dtls-cookie-exchange disabled
  return
 return

 config static-stack-settings

#
# Set the maximum number of concurrent sessions.
#
  set max-number-of-sessions 100000

#
# So we don't have to worry about matching host names in the location cache.
#
  set location-lookup-pattern user-only
 return

#
# Contains specific session configurations that are set in Signaling Engine configurations
# to be sent to Media Engine depending on the type of call
#
 config session-config-pool

#
# WebRTC to SIP call. Inbound leg ICE and encryption enabled.
#
  config entry web-to-sip
   config in-encryption
    set mode require
    set type multiple
    set priority-AES-128-CM-HMAC-SHA1-32 0
    set priority-AES-128-CM-HMAC-SHA1-80 1
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config header-settings
    config altered-body 1
     set altered-body "(.*)RTP/SAVP (.*)" "\1RTP/SAVPF \2" custom
    return
    config altered-body 33
     set altered-body "(.*)a=rtcp-fb(.*)nack pli\r\n(.*)" "\1a=rtcp-fb:* ccm fir \ba=rtcp-fb:*
nack\r\n\b\3" custom
    return
   return
   config in-ice-settings
    set admin enabled
   return
   config in-sdp-attribute-settings
    set ssrc-in-sdp strip
    set patch-audio-group enabled
   return
  return

#
# SIP to WebRTC call. Outbound leg ICE and encryption enabled.
#
  config entry sip-to-web
   config out-encryption
    set mode require
    set type multiple
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config out-media-normalization
    config codec-payload-type-bindings
     set binding telephone-event 101
    return
   return
   config out-ice-settings
    set admin enabled
    set suppress-re-invites enabled
   return
  return
```

```
#
# WebRTC to WebRTC call. Inbound and outbound ICE and encryption; enable rtcp-mux for rtp and
# rtcp to use the same port.
#
  config entry web-to-web-anchored
   config in-encryption
    set mode require
    set type multiple
    set priority-AES-128-CM-HMAC-SHA1-32 0
    set priority-AES-128-CM-HMAC-SHA1-80 1
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config out-encryption
    set mode require
    set type multiple
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config sdp-regeneration
    set origin pass
    set add-rtpmaps enabled
    set pass-attribute candidate
    set pass-attribute ice-ufrag
    set pass-attribute ice-pwd
    set pass-attribute rtcp
    set pass-attribute rtcp-mux
    set pass-attribute rtcp-fb
    set pass-attribute ssrc
    set pass-attribute remote-candidates
    set pass-attribute fingerprint
    set pass-attribute wsc-session-id
   return
   config in-ice-settings
    set admin enabled
   return
   config out-ice-settings
    set admin enabled
    set suppress-re-invites enabled
   return
   config in-sdp-attribute-settings
    set rtcp-mux enabled
   return
   config out-sdp-attribute-settings
    set rtcp-mux enabled
   return
  return

#
# WebRTC to SIP call. Media conditionally anchored; release if possible. Inbound leg ICE and
# encryption enabled; augmented ICE enabled, and rtcp mux disabled.
#
  config entry web-to-web-anchor-conditional
   config media
    set anchor enabled
    config nat-traversal
     set symmetricRTP true
    return
    set inactivity-timeout enabled "0 days 00:01:00"
    set rtp-stats enabled
    set augmented-ice enabled
   return
   config in-encryption
    set mode pass-thru
    set type multiple
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config out-encryption
    set mode pass-thru
    set type multiple
    set reuse-key enabled
    set encryption-preferences 1 DTLS
    set encryption-preferences 2 RFC_3711
   return
   config sdp-regeneration
    set origin pass
    set add-rtpmaps enabled
    set pass-attribute candidate
    set pass-attribute ice-ufrag
    set pass-attribute ice-pwd
    set pass-attribute rtcp
    set pass-attribute rtcp-mux
    set pass-attribute rtcp-fb
    set pass-attribute ssrc
    set pass-attribute remote-candidates
    set pass-attribute fingerprint
    set pass-attribute wsc-session-id
   return
   config in-ice-settings
    set admin enabled
    set delay-stun-requests enabled
   return
   config out-ice-settings
    set admin enabled
    set suppress-re-invites enabled
    set delay-stun-requests enabled
   return
   config in-sdp-attribute-settings
   return
   config out-sdp-attribute-settings
   return
  return
 return

#
# Dial plan.  Note the special type of peer server, 'web', for event-based WebRTC endpoints
# signaled via the media control API.
#
 config dial-plan
  config route WebRTC
   config condition-list
    set to-uri-condition scheme match webrtc
   return
   set priority 80
   set location-match-preferred no
   set peer web
   set request-uri-match condition-list
  return
 return
return

config external-services
return

#
# Display Oracle Communications WebRTC Session Controller Media Engine brand in the
# web management UI.
#
config preferences
 config gui-preferences
  set show-unlicensed-features false
  set channel webrtc-session-controller-media-engine
  set display-home-page-links disabled
  set display-footer disabled
 return
return

#
# Don't forget to configure users here.
#
config access
return

config features
return
```

Next Steps

After you install Media Engine, you must complete Media Engine post-installation tasks. See "WebRTC Session Controller Media Engine Post-Installation Tasks" for instructions.

9

WebRTC Session Controller Media Engine Post-Installation Tasks

This chapter provides instructions for Oracle Communications WebRTC Session Controller Media Engine (Media Engine) post-installation tasks.

Before you complete Media Engine post installation tasks, you must complete the tasks in the following chapters:

- Installing WebRTC Session Controller Signaling Engine
- Creating and Configuring a WebRTC Session Controller Signaling Engine Domain
- WebRTC Session Controller Signaling Engine Post-Installation Tasks
- Installing WebRTC Session Controller Media Engine

Overview of Media Engine Post-Installation Tasks

Before your Media Engine nodes are ready for use, you must complete the following post-installation tasks:

- Configuring Users, Permissions, and Authorization
- Deploying the Load Factor Application
- Configuring Media Engine Communication with Signaling Engine

Configuring Users, Permissions, and Authorization

For more information on configuring user, permissions, and authorization objects and properties, see *Oracle Communications OS-E System Administration Guide*.

Deploying the Load Factor Application

The following sections describe the steps you must take to deploy the Media Engine load factor application.

About the Load Factor Application

To balance request loads between Signaling Engine nodes and Media Engine nodes, you must deploy a custom load factor application on each Media Engine node. The load factor application reports an appropriate cluster based load factor to a Signaling Engine, and Signalling Engine uses that load factor to choose which Media Engine to relay requests to. Signaling Engine favors less heavily loaded instances for new requests, balancing the load among multiple Media Engine nodes.

About Load Factor Application Virtual Host Deployment Scenarios

There are two ways you can configure the load factor application virtual host:

- Host name virtual hosting
- IP virtual hosting

When configuring hostname virtual hosting, you assign an IP address to the load factor web service application, and you provide a Domain Name System (DNS) hostname for the virtual host.

When configuring IP virtual hosting, you assign an IP address to the load factor web service and another IP address to the virtual host.

> **Note:** Hostname virtual hosting is the recommended Media Engine configuration scheme.

Configuring Host Name Virtual Hosting

To configure hostname virtual hosting:

1. Navigate to the Media Engine login page:

   ```
   https://hostname
   ```

   The login page appears.

2. Enter your administration **Username** and **Password**, and click **Login**.

   The Media Engine home page appears.

3. Select the **Configuration** tab.
4. Expand the **cluster** node and select the **interface** node of the **box** you want to configure.
5. In the ip row of the configuration table, click **Add ip**.
6. Enter a **name** for the Web service interface, configure the **ip-address** information as required, and click **Create**.
7. Specify the HTTP transmission **type**, HTTP or HTTPS, as well as the Web service **port** number, and click **Create**.
8. In the virtual-host row of the configuration table, click **Add virtual-host**.
9. Enter the DNS hostname you have configured for the virtual host, make sure **admin** is set to **enabled**, and click **Create**.

Continue to "Configuring the Virtual Host web-app-config Object".

Configuring IP Name Virtual Hosting

To IP name virtual hosting:

1. Navigate to the Media Engine login page:

   ```
   https://hostname
   ```

   The login page appears.

2. Enter your administration **Username** and **Password**, and click **Login**.

   The Media Engine home page appears.

3. Select the **Configuration** tab.
4. Expand the **cluster** node and select the **interface** node of the **box** you want to configure.
5. In the ip row of the configuration table, click **Add ip**.
6. Enter a **name** for the Web service interface, configure the **ip-address** information as required, and click **Create**.
7. Select the **interface** node again.
8. In the ip row of the configuration table, click **Add ip**.
9. Enter a **name** for the Web service interface, configure the **ip-address** information as required, and click **Create**.
10. Select the **ip** object you created for the Web service, and in the web-service row of the Configuration table click **Configure**.
11. Specify the HTTP transmission **type**, HTTP or HTTPS, as well as the Web service **port** number, and click **Create**.
12. In the virtual-host row of the configuration table, click **Add virtual-host**.
13. Enter the IP address you have assigned to the virtual host, make sure **admin** is set to **enabled**, and click **Create**.

Continue to "Configuring the Virtual Host web-app-config Object".

Configuring the Virtual Host web-app-config Object

Once you have created and configured the ip objects, you must add a web-app-config object that points to the load factor Web Archive (WAR) file, to the virtual host.

To configure the virtual host's web-app-config object:

1. Log in to the Media Engine console using a secure shell (SSH), and copy **loadfactor.war** from the **/cxc/ws/**samples directory to the **/cxc_common/webapps** directory.
2. Navigate to the Media Engine login page:

   `https://`*`hostname`*

   The login page appears.
3. Enter your administration **Username** and **Password**, and click **Login**.

   The Media Engine home page appears.
4. Select the **Configuration** tab.
5. Expand the **cluster** node and select the **interface** node of the **box** you want to configure.
6. Expand the **ip** object you created, expand **web-service**, and select the **virtual-host** object.
7. In the web-app-config row of the configuration table, click **Add web-app-config**.
8. Enter the path to the load factor application, **/cxc_common/webapps/loadfactor.war** and click **Create**.
9. In the context-parameter row of the configuration table, click **Add context-parameter**.

10. Enter **meMgmtHost** for the **name**, and the DNS name or IP address of the Web service for the **value**, and click **Create**.
11. In the context-parameter row of the configuration table, click **Add context-parameter**.
12. Enter **meMgmtUsername** for the **name**, and the hostname or IP address of the Web service for the **value**, and click **Create**.

Configuring Media Engine Communication with Signaling Engine

To enable communication between WebRTC Session Controller Media Engine (Media Engine) and Signaling Engine in your WebRTC Session Controller installation you must add the Media Engines to your Signaling Engine configuration and configure the Media Engine callback which specifies the load balancer endpoint for Media Engine nodes.

Adding Media Engines to Signaling Engine

To add Media Engines to Signaling Engine:

1. Start your Signaling Engine servers if they are not already running. See "Starting the Signaling Engine Servers" for more information.
2. Navigate to the WebRTC Session Controller Signaling Engine console and log in with your administrator username and password:

   ```
   http://hostname:port/wsc-console
   ```

   Note: The default Signaling Engine console port is 7001.

3. Select the **Configuration** tab.
4. Click **Lock and Edit**.
5. In the Media Engine pane, enter the following information:
   - **User**: Enter the Media Engine administrative username.
   - **Password**: Enter the password for the administrative username.
6. Click the Add button and enter the following information:
   - **Address**: Enter the hostname or IP address of the Media Engine Node.
   - **Port**: Enter the port of the Media Engine Node.
7. Click **OK** to save the Media Engine Node. Add additional nodes as required.
8. Click **Commit** to save your changes.

For more information on the Media Engine options, see *Oracle Communications WebRTC Session Controller System Administrator's Guide*.

Configuring the Media Engine Callback

To configure the Media Engine callback:

1. Start your Signaling Engine servers if they are not already running. See "Starting the Signaling Engine Servers" for more information.

2. Navigate to the WebLogic Server administration console and log in with your administrator user name and password:

   ```
   http://hostname:port/console
   ```

   > **Note:** The default administration console port is 7001.

3. In the Domain Structure pane, expand **Environment**, and select **Servers**.
4. In the Summary of Servers pane, select the **Configuration** tab.
5. Select your Signaling Engine server from the Servers table.

   > **Note:** You will have to repeat this procedure for each Signaling Engine in a clustered environment.

6. In the Settings for the Signaling Engine, select the **Protocols** tab.
7. Select **wsc-me-callback** from the Network Channels table.
8. If you need to override the default values for **Listen Address** and **Listen Port**, uncheck **Enabled**.
9. Update the following information:
   - **Listen Address** and **Listen Port**: Enter the hostname or IP address and the port of the Signaling Engine which will serve as the primary endpoint for the Media Engine HTTP callbacks. You can only update this field if **Enabled** is unchecked.
   - **External Listen Address** and **External Listen Port**: Enter the hostname or IP address and port of the load balancer or the Signaling Engine which will serve as the backup endpoint for the Media Engine HTTP callbacks if the primary endpoint cannot be reached.
10. If you have modified the default values for **Listen Address** or **Listen port**, check **Enabled**.
11. Click **Save**.
12. Log out of the administration interface.

Configuring Media Engine Anchoring

Table 9–1 describes the media anchoring options supported by WebRTC Session Controller.

Table 9–1 WebRTC Session Controller Routing Options

Scenario	Description
Web to Web conditional anchoring	Dynamic Media Anchoring (DMA) is enabled. Browsers are allowed to stream media between each other directly. If they cannot directly reach each other (for instance if they are behind a Network Address Translation (NAT) firewall and no Traversal Using Relays around NAT (TURN) service is configured), media is relayed through Media Engine. Calls from WebRTC endpoints to and from non-WebRTC/SIP/PSTN endpoints are automatically supported.

Table 9–1 (Cont.) WebRTC Session Controller Routing Options

Scenario	Description
Web to Web forced anchoring	DMA is enabled and *all* media is routed through Media Engine. Calls from WebRTC endpoints to and from non-WebRTC/SIP/PSTN endpoints are automatically supported.
Web to Web bypassing Media Engine	DMA is disabled, and nothing is passed through Media Engine. This should only be done for diagnostic purposes.

You can modify the anchoring behavior by modifying constants in the Signaling Engine GroovyScript library.

To change the Signaling Engine to Media Engine anchoring scheme:

1. Start your Signaling Engine servers if they are not already running. See "Starting the Signaling Engine Servers" for more information.

2. Navigate to the WebRTC Session Controller Signaling Engine console and log in with your administrator user name and password:

   ```
   http://hostname:port/wsc-console
   ```

   > **Note:** The default Signaling Engine console port is 7001.

3. Select the **Script Library** tab.

4. Click **Lock and Edit**.

5. Modify the script library as required:

   - For Web to Web conditional anchoring, set:

     ```
     public static final DMA_ENABLED = true
     public static final ME_CONFIG_NAME_DMA = web-to-web-anchor-conditional
     ```

   - For Web to Web forced anchoring, set:

     ```
     public static final DMA_ENABLED = true
     public static final ME_CONFIG_NAME_DMA = web-to-web-anchored
     ```

   - To bypass Media Engine instances for diagnostic purposes:

     ```
     public static final DMA_ENABLED = false
     ```

   > **Note:** When `DMA_ENABLED` is set to false, the value of `ME_CONFIG_NAME_DMA` does not matter.

6. Click **Validate Library** to ensure you introduced no errors. Fix any errors that are reported.

7. Click **Commit** to save your changes.

Next Steps

If you have encountered any issues, see "Troubleshooting a WebRTC Session Controller Installation" for troubleshooting instructions; otherwise, you can perform additional tasks to set up your test or production system:

- Set up WebRTC Session Controller system security.

  See the discussion about setting up and managing WebRTC Session Controller security in *WebRTC Session Controller System Administrator's Guide* and *WebRTC Session Controller Security Guide*.

- Configure the WebRTC Session Controller system.

  See *WebRTC Session Controller System Administrator's Guide*.

10

Troubleshooting a WebRTC Session Controller Installation

This chapter describes how to troubleshoot Oracle Communications WebRTC Session Controller installations.

Troubleshooting a Signaling Engine Installation

The WebRTC Session Controller Signaling Engine (Signaling Engine) installer and the Domain Configuration Wizard write information to log files. You can check those log files for information about errors and actions performed during the installation.

Signaling Engine Installation Log Files

The Signaling Engine installation logs can be found at *Central_inventory_location*/**oraInventory/logs**, where *Central_inventory_location* is the directory path to the **oraInventory** directory. If you do not know the location of your Oracle Inventory directory, you can find it in the `oraInst.loc` file in the directory, **/etc/oraInst.loc**.

The following install log files are written to the log directory:

- `install`*`date-time-stamp`*`.log`

  This is the main log file.

- `install`*`date-time-stamp`*`.out`

  This log file contains the output and error streams during the installation.

- `installActions`*`date-time-stamp`*`.log`

  This file is used by the installer GUI to keep track of internal information.

- `installProfile`*`date-time-stamp`*`.log`

  This log file contains the overall statistics like time taken to complete the installation, as well as configuration, memory and CPU details.

- `oraInstall`*`date-time-stamp`*`.log`

  This log file contains the output stream of the copy session.

- `oraInstall`*`date-time-stamp`*`.err`

  This log file contains the error stream of the copy session.

Changing the Installer Logging Level

Use the `-logLevel` parameter from the command line when you start the installer. For example:

```
java -jar wsc_generic.jar -logLevel info
```

Valid value for `-logLevel` are listed below from most detailed to least detailed:

- `debug`
- `info`
- `warning`
- `error`
- `fatal`

Signaling Engine Domain Configuration Log Files

If you encounter errors when configuring a Signaling Engine domain, you can start the Fusion Middleware Configuration Wizard with the appropriate logging options.

To enable domain configuration logging, navigate to *Oracle_home*/**oracle_common/common/bin** and start `config.sh` with the `-log` and `-log_priority` options:

```
./config.sh -log=log_filename -log_priority=log_level
```

Table 10–1 describes the `-log` and `-log_priority` options.

Table 10–1 Configuration Wizard Log File Options

Parameter	Description
`-log`	Specify the location of your log file. If you specify a log file name, it is created in the same directory as the `config.sh` script unless you add a path component. Log files are otherwise created in *Oracle_home*/**logs**. Other values that can be specified with `-log` are: ■ `stdout` This writes the error message to the standard output stream. ■ `stderr` This writes the error messages to the standard error stream. ■ `disable` This disables default logging so that no log files are generated in *Oracle_home*/**logs**.
`-log_priority`	Specify the level of detail you want included in your logs. The acceptable values are listed below, from most detailed to least detailed: ■ `debug` ■ `info` ■ `warning` ■ `error` ■ `fatal`

Troubleshooting a Media Engine Installation

WebRTC Session Controller Media Engine (Media Engine) implements specialized troubleshooting commands that can be executed at the console command line interface (CLI).

For detailed troubleshooting instructions, see the discussion of Media Engine system monitoring in *Oracle Communications Application Session Controller System Operations and Troubleshooting*.

Checking Media Engine Event Logs

You can use the following CLI command to display events logged to the Media Engine's local database:

```
show event-log
```

Information in the Media Engine event log can provide details on software errors, connectivity issues, and incorrectly configured objects and properties.

Checking for Software Faults

You can use the following CLI command to display any software subsystem faults:

```
show faults
```

In conjunction with configuration files, traces and event logs, information for the show faults command can be used to troubleshoot system issues.

Checking for Hardware Issues

You can check the status of a Media Engine installation's hardware sensors using the CLI command:

```
show sensor-events
```

Likely failure points are power supplies and cooling fans.

Checking for Networking Issues

You can check Media Engine's network interface status using the CLI command:

```
show interfaces
```