Oracle® Communications
Evolved Communications Application Server

System Administrator's Guide

Release 7.0

E51262-02

August 2015

ORACLE®

Oracle Communications Evolved Communications Application Server System Administrator’s Guide, Release 7.0

E51262-02

Contents

Part I Configuring Evolved Communications Application Server

1 Configuration Overview

2 Managing the Evolved Communications Application Server

6 Managing Alarms

7 Using EDRs for Testing and Troubleshooting

Part II Reference

8 Alarms Reference

Preface

This document describes system administration tasks for Oracle Communications Evolved Communications Application Server (OCECAS).

Audience

This book is intended for system administrators who configure and manage OCECAS.

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 that have purchased support 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

OCECAS documentation is available from the Oracle Documentation website: `http://docs.oracle.com`.

Related Documents

For more information, see the following OCECAS documentation:

- *Oracle Communications Evolved Communications Application Server Release Notes*
- *Oracle Communications Evolved Communications Application Server Installation Guide*
- *Oracle Communications Evolved Communications Application Server Concepts*
- *Oracle Communications Evolved Communications Application Server Operator's Guide*
- *Oracle Communications Evolved Communications Application Server Security Guide*
- *Oracle Communications Evolved Communications Application Server Compliance Guide*
- *Oracle Communications Evolved Communications Application Server RESTful API Reference*

- *Oracle Fusion Middleware 12c Documentation Library*
- *Oracle Database Installation Guide 12c Release 1 (12.1) for Linux*
- *Oracle Database Administrator's Guide 12c Release 1 (12.1)*

Part I

Configuring Evolved Communications Application Server

This part provides administration information that is specific to Oracle Communications Evolved Communications Application Server (OCECAS). It provides a configuration overview, information about server and user entity management, OCECAS configuration procedures, and testing and troubleshooting information along with an alarms reference.

This part contains the following chapters:

1

Configuration Overview

This chapter introduces Oracle Communications Evolved Communications Application Server (OCECAS) configuration and administration.

About the Oracle WebLogic Platform

OCECAS is based on Oracle WebLogic Server. Many system-level configuration tasks are the same for both products. This part addresses the system-level configuration tasks that are unique to OCECAS. These tasks relate to network and security configuration and cluster configuration for the engine.

WebLogic Server configuration and other basic configuration tasks such as logging are addressed in the WebLogic Server documentation. This guide refers you to the WebLogic documentation for information where appropriate. For more information about WebLogic documentation, see Oracle Fusion Middleware Documentation set at

`http://docs.oracle.com/middleware/1213/index.html`.

About Oracle Communications Converged Application Server

OCECAS provides Session Initialization Protocol (SIP) servlet support using Oracle Communications Converged Application Server which itself is built upon Oracle WebLogic Server. For more information, see Oracle Communications Converged Application Server Documentation set at

`http://docs.oracle.com/cd/E49461_01/index.htm`.

Overview of OCECAS Installation

OCECAS is installed in your environment according to the deployment planned for your environment. You can use a single computer to set up the simplest installation. For production systems, OCECAS is installed on multiple physical machines. For more information, see "Understanding Installation Topologies" in *Evolved Communications Application Server Installation Guide*.

A typical deployment pipeline consists of separate session control framework (SCF) environments for testing, staging, and production. Session control framework is the name given to the runtime session-processing architecture utilized by OCECAS. Each separate environment has its own corresponding WebLogic domain, which in turn includes multiple machines. Sets of changes are moved through this pipeline of predefined environments.

About the System Environment

A complete OCECAS system environment is made up of the management domain, a user database repository (UDR) domain, the production domain, the testing domain, and the staging domain. The minimum requirements for an OCECAS deployment consist of the management, UDR, and production domains.

You can deploy the UDR in the runtime domain. For more information about domains and their port numbers, see "About the OCECAS Domains" in *Evolved Communications Application Server Installation Guide*.

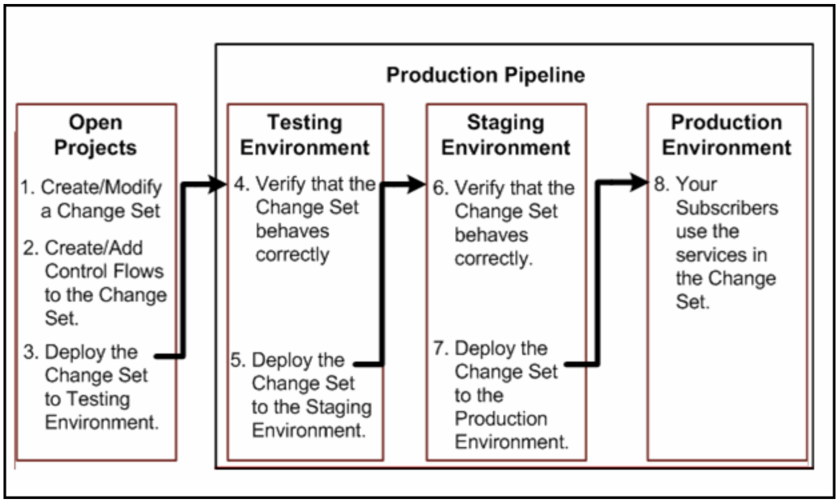
By default, the following entries are seen deployed in the Administration Console of the domains:

- Management domain: The Session Design Center as an EAR file, **oracle.occas.csp.app.sdc**.
- UDR domain: The user database repository as an EAR file, **oracle.occas.csp.app.udr**.
- Runtime domain: In each of the testing, staging, and production domains:
  - The Java application programming interface (API) for RESTful Web Services library as a WAR file, **jax-rs**.
  - The session control framework as an EAR file, **oracle.occas.csp.app.scf**.

About Change Management

Change Management is used to create and manage change as discrete blocks, from inception through to deployment. As seen in Figure 1–1, a production pipeline consists of the test, staging, and production systems.

Figure 1–1 Managing Change Sets


Production Pipeline
Open Projects
1. Create/Modify a Change Set
2. Create/Add Control Flows to the Change Set.
3. Deploy the Change Set to Testing Environment.
Testing Environment
4. Verify that the Change Set behaves correctly
5. Deploy the Change Set to the Staging Environment.
Staging Environment
6. Verify that the Change Set behaves correctly.
7. Deploy the Change Set to the Production Environment.
Production Environment
8. Your Subscribers use the services in the Change Set.

OCECAS supports segregated service data for each deployment environment, so that accidental changes on a non-production system do not impact production. You can set up environments devoted to testing, pre-production, and production. Such a scenario enables you to deploy entirely new offers with minimal effort by identifying and exporting subsets of data from one environment to another. For more about change management, see "About Change Management" in *Evolved Communications Application Server Concepts*.

About Session Design Center

The Session Design Center is the graphical user interface of the web application supported by OCECAS. It contains an interface for creating and managing control flows comprising the service logic and resources that are used to control a subscriber's voice and video sessions.

Service designers can access the Session Design Center to design their flow chart of decisions and activities for a new service. Pipelines are used to control when groups of changes are deployed as change sets and where they are deployed. For more information, see the discussion on "About the Session Design Center" in *Evolved Communications Application Server Operator's Guide*.

About Configuration and Administration Tools

You can modify the configuration of your OCECAS system, for example, by adding new servers. All OCECAS configuration and monitoring is provided through the nodes in the left pane of the Administration Console.

About Configuration Tasks

As a system administrator, you manage the configuration of both WebLogic server and OCECAS. Common configuration tasks include configuring:

- SIP container properties.
- WebLogic server network channels to handle SIP and HTTP traffic.
- OCECAS signaling properties.
- Logging servlets to record SIP requests and responses; and manage log records.

For information about the use of the WebLogic Server tools such as the Administration Console or the command-line tools, see "Overview of the Administration Console" in *Oracle Fusion Middleware Understanding Oracle WebLogic Server*.

About the OCECAS Nodes

Table 1–1 lists the nodes available to configure OCECAS:

Table 1–1 OCECAS Configuration Nodes

Node Name	Description	Domains
Evolved Communications	WebLogic Server Administration Console extension that provides access to various configuration values such as event data records (EDRs) specific to OCECAS.	Available in **sdc_management_domain**, **scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**. Not available in **scf_udr_domain**.
Diameter	Presents configuration settings and monitoring pages for the Diameter nodes and Diameter protocol applications used in the implementation.	Located in the runtime domains **scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**
Sip Server	Presents SIP Servlet container properties and other engine tier functionality. This extension also enables you to view (but not modify) SIP engines.	Located in the runtime domains **scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**

Configure these nodes in the runtime domains for the service control framework. They have extra configuration settings specific to each domain.

About Configuration Methods

You configure the domains in OCECAS by using the following tools:

- Configuration Wizard. See "About the Configuration Wizard".
- OCECAS Administration Console. See "About the OCECAS Administration Console".
- Editing the configuration files. See "About the OCECAS Configuration Files".

The methods described in the following sections can be used for certain configuration tasks.

About the Configuration Wizard

You use the Configuration Wizard to manage the domains in your OCECAS installation. For example, you can create domains, add domains, or combine two domains by creating a separate domain and adding it to an existing domain. You can also combine multiple domains into a single unit.

For information about creating OCECAS domains by using the Configuration Wizard, see "Creating Domains Using the Graphical Domain Configuration Wizard" in *Evolved Communications Application Server Installation Guide*.

About the OCECAS Administration Console

OCECAS extends the WebLogic Server Administration Console with more configuration and monitoring pages. The settings for the Administration Console interface for OCECAS are similar to the core console available in Oracle WebLogic Server.

To configure OCECAS features:

1. Ensure that your WebLogic Administration Server is running.
2. Use your browser to access the URL for the required domain:

   ```
   http://address:port/console
   ```

where *address* is the Administration Server's listen address and *port* is the listen port for the specific domain. The default port number is **7001**.

For information about domains and port numbers, see "About the OCECAS Domains" in *Evolved Communications Application Server Installation Guide*.

3. Select the node in the left pane.

   The right pane of the console provides the page or pages used for configuring and monitoring the node in the specific domain.

4. Restart the server, if necessary.

About the OCECAS Configuration Files

Table 1–2 lists the OCECAS configuration files.

> **Note:** Except for **csp.xml**, none of the configuration files should be edited manually. Perform all configuration tasks through the WebLogic Server Administration Console.

Table 1–2 Configuration Files

File	Description	Domains
approuter.xml	Part of the SIP server configuration and used to determine which application receives incoming sip messages.	WebLogic component. Available in runtime domains (**scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**
coherence-default.xml	Configuration file for in-memory data grid. Simplifies the management and deployment of Coherence clusters and Coherence-based applications.	Available in runtime domains (**scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**)
coherence.xml	Identifies servers that participate in SIP state storage, and also defines the number of threads and partitions available in the state storage service.	Available in runtime domains (**scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**)
config.xml	Specifies the name of the domain and the configuration of each server instance, cluster, resource, and service in the domain. The file includes references to more XML files that are stored in subdirectories of the *domain_home*/**config** directory. These included files are used to describe major subsystems of Oracle WebLogic Server. OCECAS custom resources use the basic domain resources defined in **config.xml**, such as network channels, cluster and server configuration, and Java Platform, Enterprise Edition (Java EE) resources.	Available in all domains

Table 1–2 (Cont.) Configuration Files

File	Description	Domains
csp.xml	Specifies the subscriber data stores (HSSs and ESS) where OCECAS retrieves subscriber data requested by applications. Also contains the federation scripts that federate, translate, and manage that data. See “About Managing and Using Subscriber Data” in *Evolved Communications Application Server Concepts* for an overview of subscriber data, and “Working With Subscriber Data” in *Evolved Communications Application Server Operator’s Guide* for instructions on how to create the federated data *views* that define this information.	Available in runtime domains
diameter.xml	Defines Diameter nodes and Diameter protocol applications used in the domain.	Available in runtime domains (**scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**)
scfDataSource-jdbc.xml	JDBC configuration file.	Available in **scf_udr_domain** and runtime domains
scfedrgeneration-jms.xml	Java Message Service (JMS) configuration file.	Not available in **scf_udr_domain**
scfqueuereplication-jms.xml	Java Message Service (JMS) configuration file.	Not available in **scf_udr_domain**
sdcDataSource-jdbc.xml	Java Database Connectivity (JDBC) configuration file.	Available in **sdc_management_domain**
sdcqueuecompiler-jms.xml	Configures the JMS Module for the compiler queue.	Available in **sdc_management_domain**
sipserver.xml	Contains general SIP container properties and engine tier configuration settings.	Available in runtime domains (**scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**)
udrDataSource-jdbc.xml	JDBC configuration file.	Available in **scf_udr_domain**

About Third-Party Software

OCECAS interacts with third-party systems for the following:

- Alarms. See "Managing Alarms".
- See "Managing Evolved Communications Application Server" for information about:
  - Web Services
  - Notifications
  - Third-party WSDLs
  - Online/offline charging
  - Media Servers
- See the documentation appropriate to your installed environment for information about:
  - Home Service Subscriber using Diameter Sh

- NoSQL
- SIP load balancing

2

Managing the Evolved Communications Application Server

This chapter introduces you to the general tasks involved in administering and managing Oracle Communications Evolved Communications Application Server (OCECAS).

For tasks specific to configuring OCECAS, see "Managing Evolved Communications Application Server".

About Administration Tasks

In your daily administration of OCECAS, you perform various tasks associated with WebLogic and other systems. The tasks include managing the following:

- Servers and Domains. See "About Managing the Servers and Domains".
- Sessions. See "About Managing Sessions".
- Connections. See "About Managing Connectivity".
- Resources. See "About Managing the System Resources".

See "Managing Evolved Communications Application Server" for information about configuration tasks specific to OCECAS.

About Managing the Servers and Domains

System administrators create domains, migrate a domain from one environment to another, and track changes in the domains.

For more information about managing servers and domains, see "Creating and Configuring OCECAS Domains" in *Evolved Communications Application Server Installation Guide.*

About Managing Sessions

Based on your installation, you use one of the following to manage sessions:

- Core Session Manager. See "About Managing Core Session Manager Elements".
- Session Border Controller. See "About Managing Session Border Controller".
- Unified Session Manager. See "About Managing Unified Session Manager Elements".

About Managing Core Session Manager Elements

Oracle Communications Core Session Manager, when integrated into your environment, provides the following functions and roles:

- Session call session control function (S-CSCF).
- Interrogating call session control function (I-CSCF).

Information about the Core Session Manager elements is available in the *Oracle Communications Core Session Manager* documentation set at Oracle Help Center:

http://docs.oracle.com.

About Managing Session Border Controller

Oracle Communications Session Border Controller, when integrated into your environment, provides the following functions and roles:

- Breakout Gateway control function (BGCF).
- Emergency call session control function (E-CSCF).
- IMS Access gateway (IMS-AGW).
- Interconnect Border control function (IBCF).
- Interrogating and Session call session control function (I/S-CSCF).
- Media Gateway control function (MGCF).
- Proxy call session control function (P-CSCF).

Information about Session Border Controller is available in the *Oracle Communications Session Border Controller* documentation at Oracle Help Center:

http://docs.oracle.com.

About Managing Unified Session Manager Elements

Oracle Communications Unified Session Manager, when integrated into your environment, provides the following functions and roles:

- Access Transfer Gateway (ATGW).
- Access Transfer control function (ATCF).
- Emergency call session control function (E-CSCF)
- IMS Access gateway (IMS-AGW).
- Interrogating call session control function (I-CSCF).
- Proxy call session control function (P-CSCF).

Information about Unified Session Manager elements is available in the *Oracle Communications Unified Session Manager* documentation set at Oracle Help Center:

http://docs.oracle.com.

About Managing Connectivity

As part of daily maintenance of network connectivity, you maintain the connections to the following elements:

- Web servers

For information about WebLogic servers, see the Oracle WebLogic Server 12*c* Release documentation set at Oracle Help Center.

- Databases

  For information about the managing the Oracle database, see the Oracle database 12*c* Release documentation set at Oracle Help Center.

- Internal data sources

  For information about the managing connectivity with data sources, see the appropriate documentation set at Oracle Help Center.

Oracle Help Center is located at

`http://docs.oracle.com`.

About Managing the System Resources

In OCECAS, a resource can be a web service, a server instance, or the Session Design Center application. It can also be an activity that takes place in your system. For example, RESTful web service API methods that allow applications and individuals access to a specific URI form an activity.

To manage OCECAS resources, you manage the following:

- Session Design Center. See "About Managing the Session Design Center".
- Templates. See "Managing the Templates".
- Databases. See "Managing Databases".

Managing Databases

You manage connections to one or more of the following databases:

- Oracle database. See "Topics for Administrators and Developers" at Oracle Help Center.
- User database repository (UDR) databases such as NoSQL, Home Subscriber Server (HSS). See the documentation for the database appropriate to your installed environment.

For information about managing the Coherence data grid, see *Oracle Fusion Middleware Developing Applications with Oracle Coherence*.

3 Managing User Entities

This chapter describes the Oracle Communications Evolved Communications Application Server (OCECAS) user entities, and the ways in which you secure and manage user access to the system resources.

About User Entities and Security Considerations

In OCECAS, a user entity can be a software element such as an application, or persons who are authorized to use the system resources. System administrators secure their system resources by exercising access control and configuring the scope of actions permitted for and with each resource.

As a system administrator or as a member of a team of system administrators, you authenticate each user entity before you permit access to the system elements. You manage the access setup to facilitate several usage scenarios, such as who has access to configure access to the resources such as control flows, restricted or barred number lists, notification definitions.

A security role, such as a security group, grants an identity to a user. A policy specifies which users, groups, or roles can access a resource under a set of conditions.

For more information about WebLogic Resource Security, see *Fusion Middleware Securing Resources Using Roles and Policies for Oracle WebLogic Server.*

`http://docs.oracle.com/cd/E24329_01/web.1211/e24421/understdg.htm#ROLES113`

About Authentication for Access to Session Design Center

OCECAS employs membership in its **EvolvedCommunicationUsers** group as an authentication requirement for accessing the Session Design Center GUI. All accounts authorized to access the Session Design Center GUI must belong to this group. For more information, see "Session Design Center GUI" in *Evolved Communications Application Server Security Guide*.

About the Central User Store

When operators give access to your system to users from multiple service providers, those user accounts can access your system. OCECAS authenticates the user names and passwords with the help of the centralized user store. This user store could be one of the following:

- An embedded WebLogic Lightweight Directory Access Protocol (LDAP) server. See "Managing Authentication with LDAP Servers".

- Oracle Identity Manager. See "Managing Authentication with Oracle Identity Manager".

About the System Administrator Tasks

As a system administrator, you manage the following aspects of user entities and data related to user accounts:

- Security roles. See "Managing Security Roles for User Entities".
- Authentication using an LDAP server. See "Managing Authentication with LDAP Servers".
- Authentication using Oracle Identity Management. See "Managing Authentication with Oracle Identity Manager".

Managing Security Roles for User Entities

The **EvolvedCommunicationUsers** group is created as part of the post-configuration task completed for the OCECAS management domain at installation time. For more information, see "Post-Configuration Tasks for Your Management Domain" in *Evolved Communications Application Server Installation Guide*.

Create users that are authorized to access Session Design Center in the OCECAS management domain. Access the administrative console for the management domain, enter the usernames and passwords in the security realm, and assign the user names to the **EvolvedCommunicationUsers** group.

For information about adding users using the administrative console, see the section "Creating Users for the SDC GUI" in *Evolved Communications Application Server Installation Guide*.

Managing Authentication with LDAP Servers

OCECAS uses the embedded WebLogic LDAP server. This server is the default security provider database for WebLogic authentication, authorization, credential mapping, and role mapping providers.

For more information, see "Managing the Embedded LDAP Server" in *Fusion Middleware Securing Oracle WebLogic Server*.

`http://docs.oracle.com/cd/E24329_01/web.1211/e24422/ldap.htm#SECMG327`

Managing Authentication with Oracle Identity Manager

When your installation uses Oracle Identity Management offerings, it can provide the following:

- Web access control
- Adaptive access control
- Identity federation and management
- User access provisioning
- Roles and authorization policies.

For more information about Oracle Identity Management, see "Oracle Fusion Middleware 12c (12.1.2) Interoperability and Compatibility" in *Oracle Fusion Middleware Interoperability and Compatibility Guide*.

4

Managing Evolved Communications Application Server

This chapter describes the configuration and management tasks specific to Oracle Communications Evolved Communications Application Server (OCECAS).

For general OCECAS administration tasks, see "Managing the Evolved Communications Application Server".

About OCECAS Configuration

As a system administrator, you configure and manage the OCECAS nodes in the domains created by your installation.

About the OCECAS Domains

An OCECAS implementation is made up of the management domain, runtime domains, and the optional UDR domains. Together, they comprise a deployment pipeline that you use to develop, stage, and finally deploy multimedia services for your subscribers to use. This chapter assumes that you have a production implementation.

For more information about OCECAS domains, see "About the OCECAS Domains" in *Evolved Communications Application Server Installation Guide*.

About the OCECAS Nodes

As a system administrator, you configure and manage the following OCECAS nodes:

- Evolved Communications

  The Evolved Communications node is a WebLogic Server Administration Console extension that provides access to various configuration values such as event data records (EDRs) specific to OCECAS.

- Diameter

  The Diameter node presents configuration settings and monitoring pages for the Diameter nodes and Diameter protocol applications used in your implementation of OCECAS.

- Sip Server

  The Sip Server node presents SIP Servlet container properties and other engine tier functionality. This extension also enables you to view (but not modify) SIP engines.

OCECAS provides Session Initialization Protocol (SIP) servlet support using Oracle Communications Converged Application Server (OCCAS) which itself is built upon Oracle WebLogic Server. For more information about SIP Server configuration, see the Oracle Communications Converged Application Server at

`http://docs.oracle.com/cd/E49461_01/index.htm`

Table 4–1 lists the domains in which the OCECAS nodes reside:

Table 4–1 OCECAS Domains and Nodes

OCECAS Domain	Evolved Communications	Diameter	SIP Server
sfc_management_domain	Yes	No	No
scf_testing_domain	Yes	Yes	Yes
scf_staging_domain	Yes	Yes	Yes
scf_production_domain	Yes	Yes	Yes
scf_udr_domain	No	No	No

About OCECAS Management Tasks

As a system administrator, you manage the following elements in OCECAS:

- Accounts. See "Setting Up Accounts for OCECAS".
- Evolved Communications node in the management domain. See "Evolved Communications in the Management Domain".
- Evolved Communications node in the runtime domains. See "Evolved Communications Node in the Runtime Domains".
- Diameter Node. See "Managing the Diameter Node".
- Session Design Center. See "About Managing the Session Design Center".
- Media Servers. See "Managing Media Servers".
- Templates. See "Managing the Templates".

Setting Up Accounts for OCECAS

OCECAS supports the following types of accounts:

- Administrative accounts authorized to access the management, UDR, and each of the runtime domains.

  These administrative accounts are created as post-installation tasks using the Administrator Account Screen of the Domain Configuration Wizard provided by the installation process. See the discussion on "Administrator Account Screen" in *Evolved Communications Application Server Installation Guide*.

- Non-administrative accounts authorized to access each of the domains (optional and as required by your installation). See "Creating Non-Administrative Accounts for OCECAS Domains".
- A primary account authorized to access the Session Design Center.

  This account belongs to the default user group **EvolvedCommunicationUsers**. It is created as part of the post-installation process. See the discussion on "Creating Users for the SDC GUI" in *Evolved Communications Application Server Installation Guide*.

- Other accounts authorized to access the Session Design Center (optional and as required by your installation)

  All accounts authorized to access the Session Design Center must belong to the **EvolvedCommunicationUsers** group. To set up these accounts, see the discussion on "Creating Users for the SDC GUI" in *Evolved Communications Application Server Installation Guide*.

In some installations, access to the respective OCECAS domains could be restricted to administrative accounts only. If this is the case, ensure that the access privilege associated with the user names and passwords for the administrative accounts does not allow them to change the schema in the installed databases.

Creating Non-Administrative Accounts for OCECAS Domains

To create a non-administrative account for an OCECAS domain:

1. Log in to the Administration Console of the OCECAS domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Security Realms**.

   The Summary of Security Realms page appears.

3. In the **Realms** table, click **myRealm**.

   The Settings for myrealm page appears.

4. Click the **Users and Groups** tab. Then, click the **Users** subtab.
5. Click **New**.

   The Create a New User page appears.

6. In the **Name** field, enter the user name for accessing the SDC GUI.
7. In the **Password** and **Confirm Password** fields, enter the password for the non-administrative user authorized to access the domain.
8. Click **OK**.
9. In the **Users** table, click the non-administrative user name that you created in step 6.

   The Settings for *UserName* page appears.

10. Click the **Groups** tab.
11. In the **Available** pane, select the group to which this user belongs.
12. Click the right arrow button to move the selected group to the **Chosen** pane.
13. Click **Save**.

Evolved Communications in the Management Domain

Access the Evolved Communications node in the Domain Structure of the management domain (**scf_management_domain**) to configure EDRs.

Important: All changes to the EDRs should be made using the Administration Console only. Changes to EDR should not be made in **csp.xml**.

About EDR Configuration Settings

Table 4–2 lists the settings to configure EDRs in the Administration Console for the management domain:

Table 4–2 EDR Settings

Entry	Description
Node Name	Name of the node to be prefixed to EDR files. The EDR files use the format, *Node Name*_-RC.date_-time.edr Where: ■ *Node Name* is the name of the node. The default node name is **ecas**. ■ RC is the running count for a day. This number starts at 1 and increased for every new file. ■ date is in YYYYMMDD format ■ time is in HHMMS format, where 'S' is in ASCCI the sign of the local time differential from UTC (+ or -) An example file name with the default node name is ecas_-_1.20140616_-_0315+1200.edr
EDR Directory Name	The directory where EDR files are placed before the files are moved to the archive location.
Maximum EDR file size	The maximum size of EDR file (1 - 2097150 kilobytes). When this limit is reached, a new file is created. The default size of an EDR file is 500 kilobytes.
File Close timeout	The duration in seconds for which an EDR file is open for writing. After that duration, EDRs are written to new file. The default is 1800 Seconds.
Days to keep in EDR directory	The Number of days an EDR file is kept in the EDR directory, After that the EDR File is moved to the Archive location. The default value is 10 days.
Archive directory location	The directory location where EDR files can be archived for longer duration.
Days to keep in Archive directory	The number of days for which the EDR files are kept in the Archive directory. After this duration, the files are deleted. The default value is 0, indicating that the files are never deleted.
Filter to exclude EDRs based on Data	An EDR which meets this filtering criteria will not be written to the EDR file. The default value is empty. All EDRs are written to the file.

Table 4–2 (Cont.) EDR Settings

Entry	Description
Configuration Cache duration	The duration (in milliseconds) for which the EDR configuration will be cached in the system. This entry makes the system efficient for getting the EDR configuration. The default value is 10000ms, indicating that the EDR configuration is cached for 10000 milliseconds. EDR configuration cache duration changes are effective when the settings are refreshed.

Configuring EDRs

Use the Administration Console to configure EDRs in the management domain, **scf_management_domain**.

To configure EDRs for the OCECAS management domain:

1. Log in to the Administration Console of the OCECAS management domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications EDR Configuration page appears.

3. Configure the EDRs. For a description of the fields, see Table 4–2.
4. Click **Save**.

Evolved Communications Node in the Runtime Domains

Use the Administration Console to configure the Evolved Communications node in each of the OCECAS runtime domains. You need to configure and manage the following:

- Alarms. See "Managing Alarms".
- Single Radio Voice Call Continuity (SRVCC). See "Managing Single Radio Voice Call Continuity".
- Statistics. See "OCECAS Statistics and System Administration".
- Telemetry. See "Managing Telemetry".
- Tracing. See "Managing Tracing".
- Conference. See "Managing the Conference Feature".
- Charging. See "Enabling Charging in the Runtime Domains".

Managing Single Radio Voice Call Continuity

Single Radio Voice Call Continuity (SRVCC) is the ability to continue a call when a subscriber moves from the long-term evolution (LTE) network (packet-switched network) to a legacy circuit-switched network. Such a switch occurs because the subscriber moves out of range of the LTE network.

OCECAS provides a Service Centralization and Continuity Application Server (SCC AS) that communicates with a Home Subscriber Server (HSS) using SIP, Interrogating Call State Call Function (I-CSCF), Serving Call Session Control Function (S-CSCF). For more information, see "About SRVCC" in *Evolved Communications Concepts*.

You can enable or disable support for SRVCC for each domain. OCECAS determines whether SRVCC-specific processing is allowed for a domain based on whether SRVCC is enabled or disabled for that domain.

It is possible to enable SRVCC in some domains, but not enabled in others. For example, you may have disabled SRVCC in all the domains. Subsequently, you may decide to enable it on a test system. In that case SRVCC would be enabled for the testing domain, and disabled for the staging and production domains.

> **Note:** If SRVCC is disabled in a runtime domain, the extra SIP headers and message flows are not included by or processed by OCECAS. If a user equipment attempts to perform call transfer (through SRVCC), OCECAS will deny the request.

SRVCC Configuration Settings

The configuration options for SRVCC are:

- **Enable SRVCC**

  SRVCC is disabled, by default. Select **Enable SRVCC** to enable it.

- **ATU STI UrI**

  A URI that is submitted to the Access Transfer Control function in a SIP message when OCECAS (acting as the SCC-AS) successfully processes a user equipment (UE) registration.

- **Transfer Timeout**

  The transfer timeout milliseconds when OCECAS (acting as the SCC-AS) processes an SRVCC Access Transfer Request.

Configuring SRVCC

To configure SRVCC for a runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications Configuration page appears.

3. Click **SRVCC**.
4. Provide the configuration settings. See "SRVCC Configuration Settings".
5. Click **Save**.

OCECAS Statistics and System Administration

OCECAS supports the gathering of statistics for particular events and enabling session designers to make decisions based on the value of the collected data. Statistics data can be gathered for events that are session-local or system-wide. OCECAS stores session-local event data within a coherence cache and records system-wide events in a database table.

An OCECAS statistic event may be one of the following:

- System-defined event

  System-defined events are mostly system-wide, stored in a database, and can be used to generate reports in the Session Design Center. Data associated with system-defined events is permanent and cannot be removed.

  The statistics associated with system-defined events is used for license-tracking purposes, such as the maximum number of requests allowed during a specific period. Other examples of system-defined events are the number of session failures determined by a particular control flow that has a counter set up, or the number of times a particular action is performed.

- User-defined event

  User-defined events are related to sessions and session-local events. Such data is stored within a coherence cache. For example, your subscribers want to collect system-wide voting scores on a telemarketing event. Or, your subscribers need to perform a session-related task such as selecting a random competition winner by connecting the N-th caller.

  Session designers generate and use session-related event statistics in OCECAS Session Design Center. They configure the **Increment Statistic** activity in a control flow to increment the session-local statistic. At a later point in the control flow, they use the resulting statistic value to configure the **Statistic Branching** activity to take decisions on the flow logic.

  For more information about how session designers retrieve statistics in OCECAS, see the discussions on "Getting Statistics Definition" and "Getting Statistics" in *Evolved Communications Application Server Operator's Guide.*

For system administrators, there is no configuration task associated with the gathering of statistics in OCECAS.

Managing Telemetry

Telemetry is the process by which you measure and collect data at various points in the communications flow and use them to monitor and manage the system. OCECAS records both system telemetry and statistics.

About System Telemetry

System Telemetry consists of recording and reporting on data generated during a session, with specific regard to the performance of particular subsystems, such as an action or an activity. For example, you can collect the time taken to:

- Load the Bootstrap control flow logic (maximum and average).
- Retrieve a UDR record (maximum and average).
- Interact with a specific web service (maximum and average).
- Traverse a segment of a control flow. See "Monitoring a Control Flow Segment".

Monitoring a Control Flow Segment

You can collect the time taken to complete a section of the control flow. For example, to see how long it takes to traverse a section of a control flow, mark the start and end of the path segment. To do so, add an extra activity or set a flag on an activity or branch and specify a name for the telemetry data item. This telemetry data item is stored in service data and associated with the control flow.

At runtime, OCECAS records telemetry for the data item for all sessions using the control flow. When a session passes through the Start Recording Message activity, it records a start time. When recording activity stops or the control flow ends, it records the stop time. OCECAS then calculates the time spent traversing the segment. It stores a summary of the results in a database table and displays the summary in the control flow editor.

How Telemetry Works in OCECAS

In OCECAS, the telemetry feature requires actions on the part of system administrators and the users of the Session Design Center. The requirements for telemetry are:

- Enabling of Telemetry in each runtime domain.

  Telemetry is **disabled**, by default. System administrators must enable telemetry in each runtime domain.

- The telemetry records.

  Each telemetry record provides the name (the telemetry record identifier), a value, and a path to the record. System administrators configure these telemetry record identifiers in the **Evolved Communications** node for the runtime domains.

  > **Important:** Use alphanumeric characters to specify the [name] attribute of an external concept. For example, *tmtryrec1*, *cflow343tmtryrec2* are valid names.
  >
  > You can use **-**, _, and blank (or space) character special characters in the name of an external concept. No other special character is allowed in the [name] entry.

  In production mode, a telemetry record is selected and called into action for an activity within a control flow of a change set.

In order to be able to collect telemetry values for a runtime domain, the following steps must be completed for the domain:

1. Service designers or operators:

   a. Determine the telemetry records they wish to use. Each telemetry record is identified by a name, for example, *tmtryrec1*, *cflow343tmtryrec2*.

   b. Provide these telemetry record identifiers to their OCECAS system administrators. For example, *tmtryrec1*, *cflow343tmtryrec2*.

2. System Administrators:

   Configure telemetry in the Administration Console of the runtime domains. See "Configuring Telemetry".

3. Service designers or operators:

   a. Access Session Design Center.

b. Provide these telemetry records as additional external concepts for the Telemetry activity. For information about providing external concepts, see the discussion on "Working with External Concepts" in *Evolved Communications Application Server Operator's Guide*.

Configuring Telemetry

At this point, you have the telemetry record identifiers (in our example, *tmtryrec1*, *cflow343tmtryrec2*) provided by the service designers or operators of Session Design Center.

To configure telemetry for a runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

   http://*hostname***:***port***/console**

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications Configuration page appears.

3. Click **Telemetry**.

4. To specify that the duration time should be added to the telemetry, select the **Enable Duration** check box.

5. In the **Telemetry Sections** field, do one of the following.

   - To enable Services Gatekeeper to recognize and accept all telemetry records entered by the session designer, enter **all**.
   - To restrict Services Gatekeeper to accepting strings pre-configured in the Administration Console, enter the strings provided by the users of Session Design Center.

     For example, if you entered *tmtryrec1*, *cflow343tmtryrec2* as the telemetry record identifiers, Services Gatekeeper uses only these two types of telemetry records.

6. Click **Save**.

At this point the telemetry records are ready to be used in the runtime domain. Update the telemetry settings for the remaining runtime domains in which the telemetry records may be called in to act on a control flow.

Also, inform the respective service designers that the telemetry records (in our example, *tmtryrec1*, *cflow343tmtryrec2*) are available for use in these domains.

Managing Tracing

Operators can use tracing in OCECAS to set up and manage the tracing rules that enable them to debug sessions. They create SIP-triggered tracing rules and service triggered tracing rules.

About Tracing Rules

Create rules for the tracing using the JSON format and save them. Each rule has a name that is used for verification. And a condition such as the name of the SIP request method. For example:

```
{
  "all_644123456" :
    {
      "method" : "INVITE",
      "P-Served-User" : ".*\\+644123456.*"
    },
  "og_644123456" :
    {
      "method" : "INVITE",
      "P-Served-User" : ".*\\+644123456.*sescase=orig.*"
    },
  "ic_644123456" :
    {
      "method" : "INVITE",
      "P-Served-User" : ".*\\+644123456.*sescase=term.*"
    },
  "644123456_to_644123457" :
    {
      "method" : "INVITE",
      "P-Served-User" : ".*\\+644123456.*sescase=orig.*",
      "To" : ".*\\+644123457.*"
    }
}
```

During runtime when the tracing level is not OFF, OCECAS creates a trace file for all sessions that match the tracing rule. The name of the trace file contains the date and time of capture, the WebLogic node, and an ID derived from the session ID to ensure uniqueness.

For example,

```
/trace/sip/og_644123456/201501211456-engine1-16b8d52353239611
```

The path name of the file indicates it contains a SIP-triggered capture for an example rule og_644123456 made at 14:56 hours on 12th January 2015 from engine1.

About Tracing Configuration Settings

Table 4–3 lists the configuration settings to configure tracing in the Administration Console for the specific runtime domains.

Table 4–3 Tracing Configurations

Name	Description
Tracing Level	Specifies the level of log events that are captured in the trace file. Set the **Tracing Level** parameter to the desired level for the specific runtime domain. For example, if you select **ERROR** in a domain, the resulting trace file in that domain contains messages for critical errors only. OCECAS supports ALL, DEBUG, ERROR, INFO, OFF, TRACE, and WARN levels. By default, the tracing level is OFF. If there are no configured rules, then the level of tracing is not changed.
Tracing Rule	Specify the rules on which Tracing is based. If there are no configured rules, then the system records no trace data. See "About Tracing Rules".
Trace Directory	The root directory for trace. Each server creates a sub directory to store trace files.

Configuring Tracing

Use the Administration Console to configure tracing for the specific runtime domains.

To configure tracing for an OCECAS runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications Configuration page appears.

3. Click the **Tracing** tab.
4. Configure the tracing for this domain. For a description of the fields, see Table 4–3.
5. Click **Save**.

Managing the Conference Feature

OCECAS manages three-way sessions as defined in Section 5.3.1.3.3 of 3GPP TS 24.147.

The Administration console of each runtime domain displays the **Conference Factory URI** option in the **Conference** tab for the **Evolved Communications** node. You can specify the URI as a regular expression allowing a number of different URIs to be valid as a factory URI.

Configuring the Conference Feature

Use the Administration Console to configure tracing for the specific runtime domains.

To configure the conference feature for an OCECAS runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications Configuration page appears.

3. Click the **Conference** tab.
4. In the **Conference Factory URI** field, enter the URI to match the initial request URI for access to the conference creation service.
5. Click **Save**.

Enabling Charging in the Runtime Domains

Charging is disabled in each of the runtime domains, by default.

To enable charging for an OCECAS runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Evolved Communications**.

   The Evolved Communications Configuration page appears.

3. Click the **Charging** tab.
4. Select the **Enable Charging** check box.
5. Click **Save**.

Managing the Diameter Node

The Diameter node is available in **scf_testing_domain**, **scf_staging_domain**, and **scf_production_domain**, the runtime domains of OCECAS. By default, transport level security (TLS) is enabled on port **3865**.

Adjust the Diameter configuration for your specific requirements, such as hosts, domains and ports.

About the Diameter Ro and Rf Interfaces

OCECAS blocks the usage of a resource until it receives authorization for the use of that resource by the requesting party. The online charging system (OCS) performs rating and balance management and approves the authorization for the use of resources. OCECAS passes the request information to the OCS through the Diameter interface (Ro) which then reserves the resource and records the usage of the resource.

The following are not supported in this release:

- Credit pooling, the ability of the server to grant the client a pool of credit from which all services draw funds several when services reserve funds against the same account
- Charging for data usage

Configuring and Managing the Ro and Rf Interfaces

By default, the installation process ensures that the Diameter Ro and RF interfaces are appropriately configured.

You can configure, enable, and manage the Ro and Rf Diameter interfaces by configuring the **virt-diameter1** node in the domain structure pane of the Administration Console for the domain.

OCECAS uses the **Maximum Request Attempts** entry as the number of times a request should be sent before treating the request as timed-out without an answer. By default, its value is **3**.

For information see the discussion on "Working with Charging Templates" in *Evolved Communications Application Server Operator's Guide*.

Configuring the Maximum Number of Attempts on a Request

To configure the **Maximum Request Attempts** for the **virt-diameter1** node in an OCECAS runtime domain:

1. Log in to the Administration Console of the OCECAS runtime domain with your administrator user name and password:

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

   where *hostname* is the IP address or name of the machine that hosts the domain and *port* is the Administration Console access port number.

2. In the Domain Structure pane on the left side, click **Diameter**.

   The Diameter Configuration Summary appears.

3. In the Diameter Configurations table, click on **virt-diameter1**.

   The Diameter Configuration Summary for **virt-diameter1** appears.

4. Click the **General** tab.

5. In the **Maximum Request Attempts:** field, enter the number of times to try a request before treating the request as timed-out without an answer.

6. Click **Save**.

About Managing the Session Design Center

Session Design Center is deployed in the management domain. It operates as a closed site with a strict requirement that the user is authenticated and authorized. The main entry points for the website are:

- **login.html:** The **login.html** provides access to the parts of the website that serve to authenticate the user.
- **index.html**: The **index.html** provides access to the parts of the application to be used by the authenticated account holder.

For more information, see the discussion on "Using the Session Design Center" in *Evolved Communications Application Server Services Guide*.

Managing the Templates

OCECAS uses the following templates:

- Session Design and Control Installation template

  The Session Design and Control Installation template installs the core functionality of the Session Design Center. The installation process completes this step. It does not provide any working service.

- VoLTE and voice and messaging over Wi-Fi (VoWiFi) Services template

About the VoLTE and VoWiFi Services Template

The VoLTE and VoWiFi Services template contains all the basic elements required to run VoLTE and VoWiFi. By default, no subscriber data is provisioned.

This template contains the following:

- All external concepts required by the default **VoLTE, VoWiFi, and eSRVCC** application. The term **eSRVCC** stands for enhanced Single Radio Voice Call Continuity.
- The default UDR View (CSP/product/app/scf/udrview)
- A default service provider. The format of the service data for this provider matches that expected by the default UDR View.
- The following schema as JSON files:
  - chargingTemplateSchema.json
  - mappingSchema.json
  - mediaResSchema.json
  - mediaSvrSchema.json
  - parametersSchema.json
  - prefixTreeSchema.json
  - serviceDataSchema.json
  - templateSchema.json
  - udrSchema.json
- Prefix Trees: The CountryCodePrefixTree.json file provides a full set of global international prefixes to ISO 3166-1 alpha-3 country codes. This file is used to obtain the country code for the called party
- Mapping Files: The CountryCodeMapping.json mapping file provides a mapping between Network ID values and ISO 3166-1 alpha-3 country codes. Network ID values could be the subscribers home network ID or obtained from the P-VISITED-NETWORK-ID header value when roaming.
- Media Server Configuration: Some control flows require a media server to play announcements. This template delivers initial service data that configures the various announcements along with a dummy driver template.

  Update the driver template to provide the real data URL or IP address for the media server. Ensure that you have fully configured the media server.

5

Managing Media Servers

This chapter describes how you can manage media servers supported by the runtime domains raised in Oracle Communications Evolved Communications Application Server (OCECAS).

About Media Servers

OCECAS does not deliver a media server. However, it organizes the media servers you install and configure and provide into groups. You set up the media resources by configuring the media server for your platform and configuring the announcement resources by specifying the correct paths to the resources.

OCECAS uses the *JSR 309 Media Control Server API Standard* to communicate with media resource servers. The JSR 309 specification is available from the Java Community Process website:

`https://jcp.org/en/jsr/detail?id=309`

How are Media and Server Resources Used

Media resources and the associated media server resources are used when call sessions support the use of media.

Service designers create the necessary announcements in the media format supported by the media server. You can store media files in the following formats:

- .m4a
- .mpg
- .mp3
- .wav

For more information, see "Working with Media Resources and Servers", see *Evolved Communications Server Operator's Guide*.

The announcement resources are grouped service data objects. For example, the music to play while the caller is on hold, or the announcement that tells the callee the session has ended could be stored under **Play**. These resources are associated with *mediaserver_name* **/MediaServer/Production/VoLTE**, where *mediaserver_name* is the name of the media server.

When included in a control flow, the **Start Playing Media** activity requires a service designer to specify the media server and the media resources to play. And if the service designer includes the Play Completed selection filter for the **Wait for Event** activity in the control flow, the software references the media server and the media resources specified for the **Start Playing Media** activity.

To support the use of the media resources, service designers configure various attributes such as the media server, the language and how long the resource must play. They can select to play the media at the initiating endpoint, destination endpoint, event source, or the conference endpoint. For example, selecting the initiating endpoint results in the announcement being played to the caller who started the session.

At runtime, the request sent to the media server is constructed from the selected endpoint, chosen language of the user, and the configuration of the selected media resource. The media file name specified for the user's language (locale) is sent in the request to the media server, which uses the file name to construct the announcement played to the user.

About the Supported Media Servers

OCECAS supports the use of the following JSR 309 compliant media servers:

- Dialogic:

  Dialogic enables service providers and application developers to elevate the performance of media-rich communications across the most advanced networks.

- Radisys:

  Radisys WebConnect is a software library that implements a JSR309 compliant interface and enables Java developers to provide real-time, multimedia processing capabilities for communication services. In the runtime environment, JSR-309 media processing requests are converted to IMS-based SIP and media server markup language commands.

If you plan to use Dialogic or Radisys media servers, configure the OCECAS to support the media server so that OCECAS can communicate with that server and play the required announcements to the parties in a call session.

Configuring OCECAS to Support Media Servers

To configure OCECAS to support your media servers, complete the following tasks for each of the runtime domains in your OCECAS installation:

1. Install the appropriate JSR 309 software for your media server. See "Installing the JSR 309 Software for Media Servers".
2. Customize the domain-wide server parameters to include the media server. See "Customizing the Media Server Parameters".
3. Update the properties files used to set up configuration for JSR 309. See "Updating the JSR 309 Properties File for Runtime Domains".
4. Deploy the Dialogic or Radisys SIP application to the engine tier cluster. See "Deploying the Driver Activator Application to the Engine Tier Cluster".

Installing the JSR 309 Software for Media Servers

Install one or both of the following software in each of the OCECAS runtime domains:

Note: OCECAS supports these two media servers only.

- Dialogic JSR 309 Connector software. See "Installing the Dialogic JSR 309 Connector Software".
- Radisys Web Connect JSR 309 Adapter software. See "Installing the Radisys WebConnect JSR 309 Adapter Software".

Important: Do not modify the **setDomainEnv.sh script** referred to by the media server installation instructions, even if the media server installation instructions asks you to do so.

Make changes to the **setUserOverides.sh script** as a customizing step. See "Customizing the Media Server Parameters" for more information.

Installing the Dialogic JSR 309 Connector Software

Install the Dialogic JSR 309 Connector software in each of the OCECAS runtime domains.

To install the Dialogic JSR 309 Connector software:

1. Download the Dialogic Power Media XNMS JSR 309 Connector software onto your platform from the following location:

   `http://www.dialogic.com`.

2. Install Dialogic XMS JSR 309 Connector software according to the instructions in the following user guide:

   `http://www.dialogic.com/webhelp/XMS/2.4/XMS_JSR309InstallConfigOCCAS.pdf`.

Installing the Radisys WebConnect JSR 309 Adapter Software

Install the Radisys Web Connect JSR 309 Adapter software in each of the OCECAS runtime domains.

Note: A valid customer login is necessary to obtain and install this software.

To install the Radisys Web Connect JSR 309 Adapter software:

1. Download the Radisys Web Connect JSR 309 Adapter software onto your platform from:

   `http://www.radisys.com`.

2. Install Radisys Web Connect JSR 309 Adapter software according to the instructions in the appropriate user guide available at

   `http://www.radisys.com/support/get-support/media-server/`

Customizing the Media Server Parameters

The **bin** directory for each runtime domain contains a file called **setUserOverrides.sh** that contains startup parameters that apply to all servers in the domain.

For each of the OCECAS runtime domains, modify the **setUserOverrides.sh** file to include the required media server.

To do so:

1. Go to the following location:

   Ocecas_home/**user_projects/domains/***domain***/bin**

   where *Ocecas_home* is the directory in which the OCECAS software is installed. And *domain* is one of the following:

   - For the testing domain: **scf_testing_domain**
   - For the staging domain: **scf_staging_domain**
   - For the production domain: **scf_production_domain**

2. Open the **setUserOverrides.sh** file in a text editor.

3. Add the following statements.

   For:

   - Dialogic media server:

     ```
     #Dialogic additions
     DLG_PROPERTY_FILE="${DOMAIN_HOME}/config/dlgc_JSR309.properties"
     export DLG_PROPERTY_FILE
     echo "DLG_PROPERTY_FILE=${DLG_PROPERTY_FILE}"
     ```

   - Radisys media server:

     ```
     #Radisys additions
     RSYS_PROPERTY_FILE=${DOMAIN_HOME}/config/rsys-connector.properties
     export RSYS_PROPERTY_FILE
     echo "RSYS_PROPERTY_FILE=${RSYS_PROPERTY_FILE}"
     ```

4. Save the file.

Updating the JSR 309 Properties File for Runtime Domains

For each of the OCECAS runtime domains, update the JSR 309 properties as appropriate. For:

- Dialogic media server, see "Updating the JSR 309 Connector Properties Files for Dialogic Media Servers".
- Radisys media server, see "Updating the WebConnect JAVA configuration File for Radisys Media Servers".

Updating the JSR 309 Connector Properties Files for Dialogic Media Servers

The **dlgc_JSR309.properties** file stores the IP address and port of SipServlet container running the JSR 309 Connector, as well as the Dialogic Power Media XMS IP address and port.

> **Note:** You need to update the **dlgc_JSR309.properties** file for each of the runtime domains.

The **dlgc_JSR309.properties** file is located under the **DlgcJSR309/properties** folder of the JSR 309 Connector distribution package you downloaded.

To update the JSR 309 Connector properties for Dialogic media servers for a runtime domain:

1. Go to the **DlgcJSR309/properties** folder at the location where you downloaded JSR 309 Connector distribution package. Locate the **dlgc_JSR309.properties** file.

2. Copy **dlgc_JSR309.properties** and paste it in the following location:

   Ocecas_home/**user_projects/domains/***domain*/**config**

   where *Ocecas_home* is the directory in which the OCECAS software is installed. And *domain* is one of the following:

   - For the testing domain: **scf_testing_domain**
   - For the staging domain: **scf_staging_domain**
   - For the production domain: **scf_production_domain**

3. In the *domain*/**config** directory, open the **dlgc_JSR309.properties** file in a text editor.

4. Set the connector's IP address. For example

   ```
   # Connector's address information
   connector.sip.address=xxx.xxx.xxx.xxx
   connector.sip.port=5080
   ```

   where, *xxx.xxx.xxx.xxx* is the IP address of the OCECAS runtime domain which listens on the given port.

5. Set the media server address. For example:

   ```
   #Media Server
   mediaserver.msType=XMS
   mediaserver.1.sip.address=xxx.xxx.xxx.xxx
   mediaserver.1.sip.port=5060
   ```

   where, *xxx.xxx.xxx.xxx* is the IP address of the media server.

6. Save the **dlgc_JSR309.properties** file in the *domain*/**config** directory.

7. Restart the managed servers in the OCECAS runtime domain.

Updating the WebConnect JAVA configuration File for Radisys Media Servers

The **rsys-connector.properties** WebConnect JAVA configuration file contains the Uniform Resource Identifier (URI) used in the **From** header in outgoing SIP requests, as well as the Radisys server IP address and port.

> **Note:** You need to update the **rsys-connector.properties** file for each of the runtime domains.

The **rsys-connector.properties** file is located in the **config** directory at the location where you downloaded the WebConnect JAVA software for Radisys media resource function.

To update the **rsys-connector.properties** WebConnect JAVA configuration file for Radisys media servers:

1. Go to the **config** folder at the location where you downloaded the WebConnect JAVA software package. Locate the **rsys-connector.properties** file.

2. Copy **rsys-connector.properties** file and paste it into the following location:

   Ocecas_home/**user_projects/domains/***domain*/**config**

   where *Ocecas_home* is the directory in which the OCECAS software is installed. And *domain* is one of the following:

- For the testing domain: **scf_testing_domain**
- For the staging domain: **scf_staging_domain**
- For the production domain: **scf_production_domain**

3. In the *domain*/**config** directory, open the **rsys-connector.properties** file in a text editor.
4. Set the FromURI value. For example:

   ```
   # URI of the WebConnect Java used in the SIP From Header
   FromURI=rsysconnector@xxx.xxx.xxx.xxx:5060
   ```

   where, *xxx.xxx.xxx.xxx* is Uniform Resource Identifier (URI) of the WebConnect JAVA used in the **From** header in outgoing SIP requests.
5. Set the media server address. For example:

   ```
   #Media Server
   mediaserver.count=1
   mediaserver.1.sip.address=xxx.xxx.xxx.xxx
   mediaserver.1.sip.port=5060
   ```

   where, *xxx.xxx.xxx.xxx* is the local IP address for the OCECAS runtime domain.

   For information about the media server parameters of the **rsys-connector.properties** file, see the *Installation and User Guide* for *WebConnect JAVA for Radisys Media Resource Function* available at:

   http://www.radisys.com/support/get-support/media-server/
6. Save the **rsys-connector.properties** file in the *domain*/**config** directory.
7. Restart the engine1 server after installing the driver software.

Deploying the Driver Activator Application to the Engine Tier Cluster

You need to deploy the appropriate (Dialogic or Radisys) driver activator application in each of the runtime domains.

To deploy the appropriate driver activator application in a runtime domain:

1. Access the Weblogic Administrator Console for the runtime domain.
2. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit**.
3. In the left pane of the Administration Console, click **Deployments**.
4. In the right pane, click **Install**.
5. Select the correct path for the files:

   Ocecas_home/**wlserver/csp/platform/applications**,

   where *Ocecas_home* is where you installed OCECAS.
6. Select the appropriate WAR file. For
   - Dialogic:

     The **oracle.occas.csp.deployable.media.dialogic.war** file.
   - Radisys:

     The **oracle.occas.csp.deployable.media.radisys.war** file.

7. When you have selected the appropriate **.war** file, click **Next**.
8. For the **Choose Targeting Style** option, select **Install this deployment as an application**.

   Click **Next**.
9. For the **Available targets for <selected driver activator>** option, select **BEA_ENGINE_TIER_CLUST**.

   Click **Next**.
10. Modify the optional settings as your require.

    Click **Finish**.

    For more information about deploying SIP applications, see the discussion on deploying a Web application at

    `https://docs.oracle.com/middleware/1213/wls/WLACH/taskhelp/web_applications/DeployWebApplications.html`

Managing and Troubleshooting Media Servers

If you encounter an issue with the following:

- Media service announcements: Check the server console log. OCECAS places the driver output in this location

  *Ocecas_home***/user_projects/domains/***domain***/servers/logs/***engine_name***.log**

  where:

  – *Ocecas_home* is the directory in which the OCECAS software is installed.

  – *domain* is **scf_testing_domain**, **scf_staging_domain**, or **scf_production_domain**, as appropriate.

  – *engine_name* is **engine1** or **engine2** as appropriate.
- Media servers: Check the appropriate user guide for the media server.

6

Managing Alarms

This chapter describes how you can manage runtime alarms raised in Oracle Communications Evolved Communications Application Server (OCECAS) by using the Simple Network Management Protocol (SNMP) service.

About Alarms and SNMP Traps

Session Design Center generates alarms during runtime processing for the following SNMP severity levels:

- Fatal
- Critical
- Warning
- Advisory
- Informational
- Clear.

These alarms are converted to SNMP Traps and sent to a trap management application. The Oracle solution for SNMP Trap Management is Oracle Enterprise Manager. OCECAS and Oracle Enterprise Manager support SNMP v3 (RFC 3411).

Standard SNMP trap managers can process the SNMP traps that OCECAS generates. For more information, see "How Enterprise Manager Supports SNMP" in *Enterprise Manager Cloud Control Administrator's Guide.*

About Alarms Configuration Options

You configure the options for OCECAS alarms through the Evolved Communications node in the Administration Console for the OCECAS runtime domains in your installation.

Caution: Configure the alarms using the OCECAS Administration Console only.

Changes that you make to alarms using any other method are temporary only. They are lost/overwritten as they are not visible in the Weblogic Administration Console.

Table 6–1 lists the general settings to configure alarms.

Table 6–1 General Settings for Alarms

Configuration Entry	Description
Store Alarms	Select this check box to store alarms received inside the platform.
Maximum Alarms Stored	The maximum number of alarms to store. The default is **0**.
Trap Generation Interval	Specify (in milliseconds) the frequency to generate SNMP traps after receiving alarms. The default frequency is **0** milliseconds, causing OCECAS to generate traps immediately after receiving an alarm.
Trap OID Prefix	Provide the trap object identifier prefix.
Trap Destination Address	Enter the destination address to which an SNMP trap is sent. It can be a host name, an IPv4, or an IPv6 address. IPv6 addresses are wrapped using '[]', for instance: [::1].
Trap Destination Port	Specify the port of the destination to which an SNMP trap is sent. The default port **162**.
Trap Timeout	Provide the timeout (in milliseconds) when sending traps. The default timeout value is **3000** milliseconds.
Trap Retries	Specify the number of times to attempt to send a trap when there is a failure in sending a trap. Default value **2**.

Table 6–2 lists the advanced settings to configure alarms.

Table 6–2 Advanced Settings for Alarms

Configuration Entry	Description
Security Level	Select the security level for the runtime OCECAS Domain. The possible settings are: ■ NOAUTH_NOPRIV This setting supports communication without authentication and privacy. The default setting. ■ AUTH_NOPRIV This setting supports communication with authentication but without privacy. The protocols used for Authentication are MD5 (message-digest algorithm) and SHA (Secure Hash Algorithm). ■ AUTH_PRIV This setting supports communication with authentication and privacy. The protocols used for Authentication are MD5 and SHA; and for Privacy, the DES (Data Encryption Standard), and AES (Advanced Encryption Standard) protocols can be used. For privacy support, install third-party privacy packages. If you plan to use a setting other than NOAUTH_NOPRIV for the security level parameter, see "Providing Custom Security Settings".
Security Username	Enter the authorised user name on the SNMP manager/receiver.
Authorization Resource ID	This entry is required to retrieve the remote user authorization password.
Privacy Resource ID	This entry is required to retrieve the remote user privacy password.

Table 6–2 (Cont.) Advanced Settings for Alarms

Configuration Entry	Description
Authentication Protocol	Select the authentication protocol from the list. The selections are: ▪ AuthMD5 MD5 Authentication protocol ▪ AuthSHA Secure Hash Authentication protocol
Privacy Protocol	Select the privacy protocol from the list. The selections are: ▪ PrivAES128 Extended encryption for Advanced Encryption Standard (AES) 128 ▪ PrivAES192 Extended encryption for Advanced Encryption Standard (AES) 192 ▪ PrivAES256 Extended encryption for Advanced Encryption Standard (AES) 256
Security Model	The version of the User-based Security Model (USM) for Simple Network Management Protocol (SNMP). The current version is **3**.

Configuring the Alarms

Configure the alarms by doing the following:

1. Access the Administration Console for the domain.
2. Select **Evolved Communications** in the Domain Structure pane.
3. Select the **Alarms** configuration tab.
4. Configure the entries displayed in the top section of the page. For a description of the fields, see Table 6–1.
5. Configure the entries displayed in the **Advanced** section of the page. For a description of the fields, see Table 6–2.
6. Click **Save**.

About SNMP Traps

OCECAS uses the SNMP4J library and the Java Management Extensions (JMX) Bean to help in the generation of SNMP traps. The SNMP traps use an application-specific object ID (OID).

You configure the OID through the WebLogic Administration Console. The OID should match the OID specified in the MIB file shipped with the OCECAS software.

Providing Custom Security Settings

If the SNMP trap manager or SNMP trap receiver in your installation does not use NOAUTH_NOPRIV, configure the required parameters for secure access. Each of the runtime domains (testing, staging, and production domains) must be configured and secured.

About the Security for Passwords

If the security level is not NOAUTH_NOPRIV, ensure that the trap client or trap generator you employ provides the required security. Wrap the password in the user-based security model (USM) for SNMP, version 3. For information about user-based security model (USM) for SNMP, version 3, see RFC3411 at `http://tools.ietf.org/html/rfc3411`.

Configuring the Required Parameters for Secure Custom Access

Complete the following steps for each runtime domain:

1. Create a new credential mapping for the domain. See "Creating New Credential Mappings".
2. Configure the alarms with the retrieved the resource IDs. See "Configuring the Alarms with the Resource IDs".

Creating New Credential Mappings

Create a new credential security mapping by doing the following:

Creating a New Credential Mapping Entry

1. Access the Administration Console for the runtime domain.
2. In the Domain Structure panel, select **Security Realms**. The Access Summary of Security realms page appears.
3. In the **Realms** table, click on the **myrealm** entry. The **Settings for myrealm** page is displayed.
4. Click on the **Credential Mapping** tab. The **Default Credential Mappings** table lists the user password credential mappings configured for this realm using Remote Resources.
5. Click **New**.

   The **Creating the Remote Resource for the Security Credential Mapping** page appears.

Creating the Remote Resource for the Security Credential Mapping

1. If you are not using the cross-domain protocol to create a credential mapping for a remote domain user, complete this set of steps:

   a. Make sure that the **Use cross-domain protocol** attribute is disabled.

      Enter information about the remote resource to be accessed using this credential mapping. This information is used to identify the remote resource.

   b. In the **Protocol** field, enter the protocol to use to reach the remote resource.

   c. If the remote resource is identified by a host name and port:

      In the **Remote Host** field, enter the host name of the remote resource.

      In the **Remote Port** field, enter the port number of the remote resource.

   d. If the remote resource is identified by a path:

      In the **Path** field, enter the path to the remote resource.

   e. In the **Method** field, enter the method on the remote resource with which this credential is used.

   f. Click **Next**.

The **Create a New Security Credential Map Entry** page appears.

g. In the **Local User** field, enter the name of the local user that you are mapping from.

   This is the WebLogic user name that will be the initiator when you want to access the remote resource using this credential mapping.

h. In the **Remote User**, enter the name of the remote user that you are mapping to.

   This is the user name that is authorized to access the resource using this credential mapping.

i. In the **Remote Password** field, remote password required by the remote resource for the remote user you specified above.

j. In the **Confirm Password** field, re-enter the password.

k. Click **Finish**.

2. Complete this step for cross-domain security:

   Create a user name and password-based credential mapping for cross-domain security:

   a. Select the **Use cross-domain protocol**.

   b. In the **Remote Domain** field, enter the name of the remote domain that needs to interact with the local domain.

   c. Click **Next**.

      The **Create a New Security Credential Map Entry** page appears.

   d. In the **Local User** field, enter the string **cross-domain**.

   e. In the **Remote User**, enter the user name configured in the remote domain that is authorized to interact with the local domain.

   f. In the **Remote Password** field, enter the password for the remote user.

   g. In the **Confirm Password** field, re-enter the password.

   h. Click **Finish**.

Configuring the Alarms with the Resource IDs

After you create the credential mappings, you will see the resource identifiers in the resource mapping records on the Credential Mappings tab. Note down the resource IDs from the resource mapping records on the Credential Mappings tab.

Next, configure the alarms by doing the following:

1. Access the Administration Console for the domain.
2. Select **Evolved Communications** in the Domain Structure pane.
3. Select the **Alarms** configuration tab.
4. Verify that the entries displayed in the top section of the page are configured. For a description of the fields, see Table 6–1.
5. Configure the entries displayed in the **Advanced** section of the page. For a description of the fields, see Table 6–2.

Note: Input the resource IDs retrieved from the Credential Mapping. For example:

```
type=<remote>, protocol=SNMP, remoteHost=localhost, remotePort=162,
method=auth
```

7

Using EDRs for Testing and Troubleshooting

This chapter explains how to find and use the event data records (EDRs) that Oracle Communications Evolved Application Server (OCECAS) generates. This chapter also provides a table of the keys that represent activities in the EDRs. This information in this chapter is useful for developing and debugging SIP traffic.

Understanding EDRS

EDRs record details of events that occur while OCECAS is processing traffic or performing management operations. Unless you specifically filter them out, EDRs are generated for every service performed, including all voice, IFR, WebRTC, or data calls. EDRs are also generated for OCECAS management actions, interactions with applications, and with network nodes. EDRs are stored in files that have a configurable life cycle.

EDRs are stored in the OCECAS Management System domain in *Middleware_home*/**occas/sdc-root/sdc/sdc_1/user_projects/domains/sdc_management_domain/servers/mgmt1/edr**

EDR files that are open for writing use this syntax: *NodeID***.open**, for example **testing3.open**. Here, *NodeID* is the hostname of the system running the ECAS implementation.

Closed EDR filenames use this syntax:

*environmentID***_-_***running_count***.***YYYYMMDD***_-_***time***+***UTC_offset*

Where:

- *envirnmentID* is the OCECAS component generating the EDRS.
- *running_count* is the number of EDR files generated. The count starts with 1 and increments.
- *YYYYMDD* is the date format.
- *time* is the local time.
- *UTC_offset* is the number of time zones from UTC.

For example, this EDR filename: **ecas_-_1.20150616_-_0315+1200** is a closed EDR file generated by the **ecas** node. The **1** indicates that it is the first EDR file generated; the date indicates that the file was closed on June 16th, 2015 at 03:14 in the time zone 12 hours ahead of UTC.

Understanding Management Node EDRs

This example EDR is generated by creating a control flow:

```
web.URL=/api/change-sets/2/control-flows/pid/11/diffs|pfm.dom=sdc_management_
domain|web.mth=POST|web.rht=dhcp-uk-IP_address.uk.oracle.com|web.usr=user|
web.res=200|web.rqt=2015-03-02T15:08:20.967Z|
```

Table 7–1 list the important management node EDR fields.

Table 7–1 EDR Platform Identification Fields.

EDR Field	Parent PID	Added By	Name	Description	Examples
pfm.dom	200.1	Evolved Communications	Domain Name	The domain that generated the EDR. Used to filter EDRs generated from various domains.	scf_staging_domain, scf_production_domain
pfm.srv	200.2	Evolved Communications	Server Name	The name of the managed server.	engine1, engine2, mgmt1

Understanding Runtime Action EDRs

A SIP session call generated this example URD:

```
pfm.srv=engine1|pfm.dom=scf_testing_domain|call.relc=400|udr.pvd=MVNO1|
call.spmt=REGISTER|chs.cfn=SIP REGISTER|call.sedt=2015-03-02T15:02:39.960Z|
call.cid=5c72cec76eb29c78c4eb5f01abe878e3@127.0.0.1|chs.dep=21|call.pcl=SIP|
chs.tfn=ST-1-1-27,INCS-216-1-37,STRE-55-1-65,COPY-2-2-77,COPY-4-2-88,
COPY-6-1-104,COMP-135-3-115,COMP-9-1-125,COPY-76-1-144,COPY-109-1-165,
COPY-88-1-180,COMP-38-1-191,STRE-39-1-197,DTO-37-1-212,COMP-60-1-223,
COPY-61-1-240,COPY-215-2-253,CPDT-22-3-385,STRE-24-1-389,
COPY-27-1-399,COMP-106-2-407,COPY-31-1-419,COMP-65-1-427,COMP-62-1-458,
EXCT-69-1-469,EXCT-70-1-480,CPRL-66-3-497,STRE-68-1-501,AEDR-85-1-511,
TSST-72-2-564,REL-203-1-571,END-204|call.ssdt=2015-03-02T15:02:39.270Z|
call.roam=true|chs.cst=3|
```

Runtime EDRs record the activities that the SIP session traverses during session processing. The activities are referenced using the Activity *fast keys*. Fast keys are three and four-letter abbreviations that represent individual activities. Table 7–5 lists the fast keys.

Table 7–2 lists the Runtime EDR node fields.

Table 7–2 Runtime EDR Fields

EDR Field	Parent PID	Added By	Name	Description	Examples
call.ssdt	100.1	Evolved Communications	Start Datetime	UTC Datetime when service is triggered on the environment. This is not the call establishment time and you should not use it for billing purposes.	Date time in YYYY-mm-hhThh:mm:ss.SSS format. For example, 2015-09-08T20:55:09.006Z
call.dg	100.2	Evolved Communications	Calling URI	The SIP message calling URI. Derived from the /CallInfo/CallingParty /Uri in context fields.	sip:alice@atlanta.com, tel:+358-555-1234567
call.cld	100.3	Evolved Communications	Calling URI	The SIP message called URI. Derived from the /CallInfo/CallingParty /Uri context fields.	sip:bob@biloxy.com, tel:+358-555-1234568

Table 7–2 (Cont.) Runtime EDR Fields

EDR Field	Parent PID	Added By	Name	Description	Examples
call.rcld	100.4	Evolved Communications	Redirected Called URI	When Evolved Communications redirects the original called URI to another URI, then this EDR field contains the redirected URI. The **cld** EDR Field contains the Original Called URI.	sip:bob@biloxy.com, tel:+358-555-1234568
call.ctyp	100.5	An activity	Call Type	An informal description of the call type defined in the control flow.	origination, termination, and so on.
call.b2br	100.6	Evolved Communications	B2B Session Result	The result of B2B session establishment, coded as an integer value: 1 - Session established 2 - Busy 3 - No Answer 4 - Not Reachable 5 - Redirected 6 - Party A Hung Up 7 - Internal Error	1
call.b2st	100.7	Evolved Communications	B2B Session Start Time	The UTC datetime set when the B2B session was established.	Date time in YYYY-mm-hhThh:mm:ss.SSS format. For example: 2015-09-08T20:58:19.046Z
call.b2et	100.8	Evolved Communications	B2B Session End Time	The UTC datetime set when the B2B session ended.	Date time in YYYY-mm-hhThh:mm:ss.SSS format. For example: 2015-09-08T20:58:40.012Z
call.dcld	100.10	Evolved Communications	Diverted Call URI	Only present if a service rule diverts the call.	sip:bob@biloxy.com
call.drul	100.11	Evolved Communications	Call Diversion Rule	Identifies the service rule that diverted the call.	The options are: ▪ CFB - Communication Forward on Busy ▪ CFU - Communication Forward Unconditional ▪ CFNL - Communication Forward on Not Logged In ▪ CFNR - Communication Forward No Reply ▪ CFNR - Communication Forward Not Reachable
call.relc	100.12	Evolved Communications	SIP Release Cause	The SIP cause value that releases the calling party	200 (Normal Release) 400 (Bad Request) 500 (Internal Error) 480 (Temporarily Unavailable), and so on. Refer to the SIP RFC for all possible values and meanings.

Table 7–2 (Cont.) Runtime EDR Fields

EDR Field	Parent PID	Added By	Name	Description	Examples
call.spmt	100.14	Evolved Communications	SIP Initiation Method	SIP method that initiated the session	INVITE, REGISTER, and so on.
call.sedt	100.15	Evolved Communications	End Datetime	The UTC datetime that the service ended on the environment	Date time in YYYY-mm-hhThh:mm:ss.SSS format, For example: 2015-09-08T20:58:49.210Z
call.cid	100.16	Evolved Communications	SIP Call ID	The SIP ID used to identify the call	a84b4c76e66710@pc33.atlanta.com
call.pcl	100.17	Evolved Communications	Protocol	The protocol that initiated the session	SIP, Diameter, and so on.
call.bar	100.18	Evolved Communications	Call Barred	A binary value that indicates whether the call has been barred	True/False
call.roam	100.19	An activity	Call Roaming	A binary value that indicates whether the call is a roaming call	True/False
call.dtry	100.20	An activity	Country Code	The ISO 3166-1 alpha 3 country code of country that subscriber is registered in	GBR, USA, NZL and so on.
call.rgstn	100.21	An activity	Registration	The registration operation	reg, rereg, or dereg
chs.cst	500.1	Evolved Communications	Change Set ID	The change Set ID used to execute the flow	An integer for Change Set Id.
chs.cfn	500.2	Evolved Communications	Control Flow Names	The names of control flows executed, including bootstrap flow name	SIP INVITE, Session Origination, and so on.
chs.tfn	500.3	Evolved Communications	Activities traversed	Lists the activities traversed during flow execution	ST-1-1-17,FILL-5-1-30,FILL-6-1-45,END-4. See Table 7–5 for a list of the activity fast keys.
chs.dep	500.4	Evolved Communications	Deployment ID	The deployment ID used during flow execution	An integer for Deployment ID

Understanding UDR Node EDRs

This example EDR was generated by provisioning a subscriber:

```
web.URL=/api/subscriber|pfm.dom=scf_udr_domain|web.mth=POST|
web.rht=localhost.localdomain|web.usr=user|web.res=201|
web.rqt=2015-03-02T15:09:05.166Z|
```

Table 7–3 lists the EDR fields for the UDR Node.

Table 7–3 EDR Fields for the UDR Node

EDR Field	Parent PID	Added By	Name	Description	Examples
web.mth	300.1	Evolved Communications	Web Request Method	The RESTful request method name. GET requests do not generate EDRs.	POST, PUT, DELETE
web.URL	300.2	Evolved Communications	Web Request URL	This URL is relative to the document root; not the full URL.	/api/change-sets/214/service-data
web.usr	300.3	Evolved Communications	User Name	The login name of the user.	test, and so on.
web.rht	300.4	Evolved Communications	Remote Host	The name of the remote Host that sent HTTP request to Evolved Communications. This field is taken from HTTP request header.	*localhost.localdomain*
web.rqt	300.5	Evolved Communications	Request Time	The UTC datetime when the request arrived at Evolved Communications.	Date time in YYYY-mm-hhThh:mm:ss.SSS format, For example: 2015-09-08T20:58:49.210Z
web.res	300.6	Evolved Communications	HTTP Response Code	HTTP Response Code	200 (Success), 404 (Not Found), and so on. Refer to HTTP response codes for full list of values

Table 7–3 (Cont.) EDR Fields for the UDR Node

EDR Field	Parent PID	Added By	Name	Description	Examples
udr.pvd	600.1	Evolved Communications	Service Provider	The service provider name.	For example, MVNO1
udr.eec	600.2	Evolved Communications	UDR error code	UDR error code.	An integer representing the UDR error code. These are the same as HTTP status codes: ■ 2xx - Success or partial success message. ■ 4xx - Client side error message ■ 5xx - Server side error message The error reasons are listed in the udr.eer field.
udr.eer	600.3	Evolved Communications	UDR Error reason	The UDR action error or status message.	UDR error or status message corresponding to the error code in the udr.eec field: ■ 200 - Successfully read profile ■ 204 - Profile updated ■ 204 - Notification processed ■ 404 - User not provisioned ■ 500 - Error processing READ request ■ 500 - Error processing UPDATE request ■ 500 - Error processing NOTIFICATION request ■ 500 - Provider returned NULL response ■ 503 - Service temporarily unavailable ■ 504 Provider request timed out

Configuring EDR Files

You configure EDR file generation using the OCECAS Administration console.

> **Note:** You must be in Production Mode to change the EDR file configuration settings.

To configure EDR file generation:

1. Start the Administration Console.

   See "About the OCECAS Administration Console" for details.

2. In the **Domain Structure** panel on the upper left, select **Evolved Communications**.

The Evolved Communications Configuration page appears.

3. Click **Lock and Edit**.
4. Make the EDR file configuration changes that your implementation requires.

   See Table 7–4 for the list of configuration parameters, their default values, and descriptions of their behavior.
5. Click **Save** to save your changes.

You have these options for configuring EDR file generation:

Table 7–4 EDR File Configuration Parameters

EDR File Parameter	Default Value	Description
Node Name	ecas	The informal name of the OCECAS node generating EDRs.
Location of EDR Directory	*domain_home***/sdc_management_domain/servers/mgmt1/edr**	Directory where EDR files are stored before they are moved to archive location.
Maximum EDR file size:	500	Maximum size (in kilobytes) of EDR file. When the file reaches this size, it is saved and another file is created.
File Close Timeout	1800	Duration of the file life seconds. After reaching this limit the file is closed and saved, and the next EDR file is opened for writing.
Days to keep in EDR directory	10	The time limit in days to keep the EDR file in the EDR directory. After this time expires, the EDR File is moved to the Archive location.
Archive directory location:	*domain_home***/sdc_management_domain/servers/mgmt1/edr/archive**	Directory location where EDR files are archived for storage.
Days to keep in Archive directory:	0	Number of days to keep EDR files in Archive directory, After this time limit the files are deleted. The default value (0) never deletes the files.
Advanced - Filter to exclude EDRs based on Data	NA	EDRs that meet the filtering criteria are not written to EDR file. Default value is empty, so all EDRs are written to file. For example if you enter'**.*web.mth=POST.***' for filter criteria, EDRs with the web request method of POST are not stored.

Table 7–4 (Cont.) EDR File Configuration Parameters

EDR File Parameter	Default Value	Description
Advanced - Configuration Cache Duration	10000	Duration (in milliseconds) for which the EDR configuration is cached. This parameter determines how often the EDR configuration is read from the Oracle database by the management domain. Reading the database less often improves the EDR processing speed. The default value is 10000ms, that means the EDR configuration is cached for 10000 milliseconds. Note that EDR configuration changes are effective after they are refreshed.

EDR Fast Key Reference

The Session Design Center UI supports a number of configurable *activities* that you use to build SIP calls in control flows and define SIP session behavior. Each activity has a *fast key* abbreviation that identifies in EDRs. Table 7–5 lists the fast key for each activity.

Table 7–5 Fast Key Reference

Activity Name	Activity Fast Key
Adjust Media	ADJM
Add EDR Field Value	AEDR
Alarm	ALM
Array Index	AIDX
Compare	COMP
Compare Data Time	CPDT
Compare Day of Week	CPDW
Compare List and Value	CPRL
Copy Value	COPY
Date Time Offset	DTO
End	END
End Charging Session	ECS
Event Charge	EVTC
Extract and Store String	EXCT
Find and Replace and Store Value	FIND
Generate and Document and Store	DGOC
Increment Statistic	INCS
Load Service Definition	LSD
Notes	NOTE
Prefix Tree Lookup	PTLU
Release	REL

Table 7–5 (Cont.) Fast Key Reference

Activity Name	Activity Fast Key
Remote Copy	RCPY
Retrieve Session List	RSL
Route	RTE
Route Changed	RTEC
Run Control Flow	RCF
Run Service Definition	RSD
Run Web Service	RWS
Send Message	SEND
Start	ST
Start Back to Back Session	SBBS
Start Charging Session	SCS
Start Collecting Digits	SCOL
Start Conference Session	CONF
Start Playing Media	SPLY
Start Recording Message	SREC
Statistics Branching	STB
Stop Media	STPM
Store	STRE
Store Session Key	STSK
Sync Statistic	SYNS
Telemetry	TELM
Translate and Store Value	TSST
Update Charging Session	UCS
Update Profile	UPDP
Wait For Event	WFEV

Part II

Reference

This part provides reference information about Oracle Communications Evolved Communications Application Server (OCECAS) alarms.

This part contains the following chapter:

- Chapter 8, "Alarms Reference"

8

Alarms Reference

This chapter lists the various types of alarms that Oracle Communications Evolved Communications Application Server (OCECAS) generates.

Action Alarms

Table 8–1 lists the alarms generated by the OCECAS application server subsystem on the Runtime nodes. The alarms are generated mainly for general session processing for all supported session protocols such as SIP, DIAMETER, SOAP and REST.

Table 8–1 Action Alarms

Alarm	Severity	Text	Cause	Resolution
SAS.-236841626	CRITICAL	Session in state: {} so cannot send immediate response: {} ({})	Internal inconsistency	Report software fault
SAS.1358467794	CRITICAL	Failed to find SIP session so cannot send SDP	Internal inconsistency	Report software fault
SAS.1774975313	CRITICAL	UAS Session in TERMINATED state so cannot send SDP	Internal inconsistency	Report software fault
SAS.124500320	CRITICAL	Invalid SIP session and/or endpoint state	Internal inconsistency	Report software fault
SAW.807247288	CRITICAL	Web service bean failed to send request	Problem sending web request	Check logs and network
SAW.438621975	CRITICAL	SOAP action failed to find SOAP transaction in Registry	Internal inconsistency	Report software fault
SAS.-783315185	CRITICAL	B2B Session in TERMINATED state so cannot send SDP	Internal inconsistency	Report software fault
SAW.-2003283391	CRITICAL	Web service bean failed to send request	Problem sending web request	Check logs and network
SAW.-305625600	CRITICAL	REST action failed to find REST transaction in Registry	Internal inconsistency	Report software fault

Charging Alarms

Table 8–2 lists the alarms generated by the OCECAS application server Charging subsystem on the Runtime nodes.

Table 8–2 Charging Alarms

Alarm	Severity	Text	Cause	Resolution
SC.461019824	CRITICAL	Cannot send {} CCR because there is no RoSession established	Internal inconsistency	Check charging activities in control flow so report software fault
SC.-610384092	CRITICAL	OfflineCharging service: received invalid transaction	Internal inconsistency	Report software fault
SC.1438229309	CRITICAL	Could not send CCR: {}	Problem sending charging request	Check logs and network
SC.1959791603	CRITICAL	Cannot send {} because RoSession {} is already in progress	Internal inconsistency	Check charging activities in control flow so report software fault
SC.-1759787268	CRITICAL	Invalid ACR built from template: {}	A charging template was configured incorrectly	Retract the change set or fix charging template in a new change set
SC.-1954822780	CRITICAL	Failed to lookup off-line charging service '{}'	Invalid charging service configuration	Configure the charging service correctly
SC.-510513429	CRITICAL	Cannot send {} ACR because RfSession {} is in terminated state	Internal inconsistency	Check charging activities in control flow so report software fault
SC.-1378210684	CRITICAL	OnlineCharging service: received invalid transaction	Internal inconsistency	Report software fault
SC.1545207092	CRITICAL	Cannot send {} CCR because RoSession {} is in terminated state	Internal inconsistency	Check charging activities in control flow so report software fault
SC.-2106093557	CRITICAL	Could not send ACR: {}	Problem sending charging request	Check logs and network
SC.-449017300	CRITICAL	Cannot send {} because RfSession {} is already in progress	Internal inconsistency	Check charging activities in control flow so report software fault
SC.103795138	CRITICAL	Could not get diameter node from com.bea.wcp.diameter.Node attribute	Invalid diameter configuration	Check that a diameter network-access-point is defined
SC.1191760828	CRITICAL	Failed to populate charging template: {}	A charging template was configured incorrectly	Retract the change set or fix charging template in a new change set

Table 8–2 (Cont.) Charging Alarms

Alarm	Severity	Text	Cause	Resolution
SC.1477636007	CRITICAL	Cannot send {} ACR because there is no RfSession established	Internal inconsistency	Check charging activities in control flow so report software fault
SC.1932756270	CRITICAL	Invalid CCR built from template: {}	A charging template was configured incorrectly	Retract the change set or fix charging template in a new change set
SC.1873110642	CRITICAL	Could not get diameter node from com.bea.wcp.diameter.Node attribute	Invalid diameter configuration	Check that a Diameter network-access-point is defined
SC.-284808798	CRITICAL	Could not create an Online charging application	Invalid diameter configuration	Check Diameter configuration
SC.-632290436	CRITICAL	Failed to lookup on-line charging service '{}'	Invalid charging service configuration	Configure the charging service correctly
SC.-1480789177	CRITICAL	Could not send Diameter message: {}	Problem sending charging request	Check logs and network
SC.-1333890964	CRITICAL	Failed to populate charging template: {}	A charging template was configured incorrectly	Retract the change set or fix charging template in a new change set
SC.288526327	CRITICAL	Could not send Diameter message: {}	Problem sending charging request	Check logs and network

Chassis Alarms

Table 8–3 lists the alarms lists the alarms generated by the OCECAS application server subsystem on the Runtime nodes. The alarms are generated mainly for general session processing for all supported session protocols such as SIP, DIAMETER, SOAP and REST.

Table 8–3 Chassis Alarms

Alarm	Severity	Text	Cause	Resolution
SC.2119666489	CRITICAL	Invalid engine state ID {} in response	Internal inconsistency	Report software fault
SE.-259869001	CRITICAL	Invalid configuration in control flow {} for activity {} and message: {}	A control flow activity was configured incorrectly	Retract the change set or fix control flow in a new change set
SC.-1647253768	CRITICAL	Failed to load control flow key for protocol: '{}' and method: '{}'	There is no application trigger configured for the specified protocol and method	Ensure that the database contains a valid application trigger entry

Media Alarms

Table 8–4 lists the alarms generated by the OCECAS application server Media subsystem on the Runtime nodes.

Table 8–4 Media Alarms

Alarm	Severity	Text	Cause	Resolution
SM.-1340286052	CRITICAL	Failed to initialise MediaParticipant: {}	Can't process media transaction	Check the media server is available
SM.917837316	CRITICAL	Invalid media server configuration: {}	Can't process media transaction	Configure the media server correctly

User Profile Alarms

Table 8–5 lists the alarms generated by the OCECAS application server UDR subsystem on the Runtime nodes.

Table 8–5 User Profile Alarms

Alarm	Severity	Text	Cause	Resolution
SA.73957577	CRITICAL	Failed to update UDR	UpdateProfile failed	Check log for more information
SU.-852056064	CRITICAL	handleProfileNotification: session has null transaction	Internal inconsistency	Report software fault
SU.-1967106361	CRITICAL	Failed to create UDR update request	Internal inconsistency	Report software fault
SU.-756848176	CRITICAL	Failed to send UDR read request: {}	Problem sending UDR request	Check logs and network
SU.1546249155	CRITICAL	Failed to send UDR update request: {}	Problem sending UDR request	Check logs and network
SU.-561858002	CRITICAL	UDR handler '{}' returned null service	Invalid UDR service configuration	Configure the UDR service correctly
SU.-1400801029	CRITICAL	Failed to create UDR request builder	Internal inconsistency	Report software fault
SU.-208267932	CRITICAL	handleUpdateResponse: called with null response	Internal inconsistency	Report software fault
SU.837283091	CRITICAL	handleReadResponse: session has null transaction	Internal inconsistency	Report software fault
SU.1109342659	CRITICAL	handleUpdateResponse: response has null session	Internal inconsistency	Report software fault
SU.-29369338	CRITICAL	handleUpdateResponse: session has null transaction	Internal inconsistency	Report software fault
SU.2053540822	CRITICAL	handleReadResponse: response has null session	Internal inconsistency	Report software fault
SU.1699875721	CRITICAL	handleReadResponse: transaction has null app session id	Internal inconsistency	Report software fault
SU.1517428308	CRITICAL	Failed to create UDR read request	Internal inconsistency	Report software fault

Table 8–5 (Cont.) User Profile Alarms

Alarm	Severity	Text	Cause	Resolution
SU.1670068532	CRITICAL	handleProfileNotification: event has null session	Internal inconsistency	Report software fault
SU.735930231	CRITICAL	handleReadResponse: called with null response	Internal inconsistency	Report software fault
SU.397693795	CRITICAL	handleProfileNotification: called with null event	Internal inconsistency	Report software fault

Web Request Alarms

Table 8–6 lists the alarms generated by the OCECAS application server Web Service subsystem on the Runtime nodes.

Table 8–6 Web Request Alarms

Alarm	Severity	Text	Cause	Resolution
SW.2044210026	CRITICAL	Failed to find the transaction.	Internal inconsistency	Report software fault
SW.2035879783	CRITICAL	Unable to Construct REST client. REST client actions will be unavailable.	Problem sending web request	Check logs and network
SW.609982854	CRITICAL	Unable to perform REST client action because there is no REST client.	Problem sending web request	Check logs and network
SW.153863372	CRITICAL	Web session: received invalid transaction	Internal inconsistency	Report software Check
SW.1380968501	CRITICAL	Web session: received invalid transaction	Internal inconsistency	Report software fault
SW.-335797324	CRITICAL	Received REST Error response: {}	Web request failed	Check logs and network
SW.-110818330	CRITICAL	Invalid WS response context so ignoring JAX-WS response.	Internal inconsistency	Report software fault
SW.-650174813	CRITICAL	Web handler '{}' returned null service	Invalid web service configuration	Configure the web service correctly
SW.-27831726	CRITICAL	Web handler '{}' returned null service2	Invalid web service configuration	Configure the web service correctly
SW.-1023652141	CRITICAL	Failed to find the transaction.	Internal inconsistency	Report software fault