

Oracle® Fusion Middleware

Getting Started with Oracle Edge Analytics

12c Release (12.1.3)

E42054-05

November 2015

How to get started with developing applications on Oracle Edge Analytics. This document includes installation instructions. Oracle Java Embedded is a platform for developing high-performance, event-driven embedded applications.

Copyright © 2007, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Madhubala Ponnekanti, Oracle® Corporation,

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	v
Audience	v
Related Documents.....	v
Conventions.....	vi
What's New in This Guide.....	vii

1 Overview

1.1 Introduction.....	1-1
1.1.1 Use Cases.....	1-2
1.1.2 Features.....	1-3
1.1.3 Platform Support.....	1-5
1.1.4 Supported Java Versions	1-6
1.2 PreLoading and PostLoading	1-7
1.3 Application and Development Framework.....	1-8
1.4 Creating Domains and Configuring the Server.....	1-9
1.4.1 Run the Configuration Wizard in Silent Mode.....	1-9
1.4.2 Create a silent.xml File.....	1-10
1.5 Applications Development.....	1-12
1.5.1 Example Embedded Application	1-13
1.5.2 REST Web Services Support	1-13
1.6 Application Deployment	1-14
1.7 Application Administration.....	1-15
1.7.1 The policygen Command-Line Utility	1-16
1.8 Supported Platform and Resource Configurations	1-16

2 Installation

2.1 Installation Overview.....	2-1
2.1.1 Before You Install	2-1
2.1.2 Directory Structure and Concepts	2-1
2.2 Install from the Command Line	2-2

2.3 Post-Installation Steps	2-3
-----------------------------------	-----

Preface

This document provides introductory information about Oracle Edge Analytics.

Audience

This document is intended for users interested in learning about Oracle Edge Analytics. Readers should be familiar with basic Java development. Some knowledge of SQL would be helpful.

Related Documents

For more information, see the following documents in the Oracle Event Processing documentation set:

- *Getting Started with Oracle Event Processing*
- *Developing Applications for Oracle Event Processing*
- *Administering Oracle Event Processing*
- *Using Visualizer for Oracle Event Processing*
- *Oracle CQL Language Reference for Oracle Event Processing*
- *Schema Reference for Oracle Event Processing*
- *Developing Applications with Oracle CQL Data Cartridges*
- *Customizing Oracle Event Processing*
- *Java API Reference for Oracle Event Processing*
- *Java API Reference for Oracle Edge Analytics*
- *Using Oracle Stream Explorer*
- *Getting Started with Oracle Stream Explorer*
- *Oracle Database SQL Language Reference* at: http://docs.oracle.com/cd/E16655_01/server.121/e17209/toc.htm
- SQL99 Specifications (ISO/IEC 9075-1:1999, ISO/IEC 9075-2:1999, ISO/IEC 9075-3:1999, and ISO/IEC 9075-4:1999)
- Oracle Event Processing Forum: <http://forums.oracle.com/forums/forum.jspa?forumID=820>.

Conventions

The following text conventions are used in this document:

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

What's New in This Guide

This guide has been updated for the 12c Release. The examples use Oracle JDeveloper for the IDE.

Documentation for a new feature named PreLoading and PostLoading has been added to this guide in the 12c release.

Oracle Event Processing on Oracle Embedded Systems has been renamed to *Oracle Edge Analytics*.

Overview

Oracle Edge Analytics is an event processing server designed to support event processing applications in embedded environments such as those supported by the Java Embedded Suite (JES). Oracle Edge Analytics features represent a subset of Oracle Event Processing features.

The dependency on Java Embedded Suite (JES) is eliminated by pre-packaging derby and jersey libraries with Oracle Edge Analytics.

This guide introduces Oracle Edge Analytics and provides installation information. For more information about Oracle Event Processing, see *Getting Started with Oracle Event Processing*.

This chapter includes the following sections:

- [Introduction](#)
- [PreLoading and PostLoading](#)
- [Application and Development Framework](#)
- [Creating Domains and Configuring the Server](#)
- [Applications Development](#)
- [Application Deployment](#)
- [Application Administration](#)
- [Supported Platform and Resource Configurations](#) .

1.1 Introduction

Event processing applications receive potentially large amounts of streaming data that represents events and responds in real time based on the event data. Oracle Edge Analytics supports applications deployed in embedded environments often found at or near event sources. These environments include sensors such as for environment conditions and moving sources such as vehicles or mobile devices.

By deploying event processing applications in these locations, you can filter events at or near the event source, reducing the amount of network traffic flowing through other server resources, including Oracle Event Processing applications on enterprise servers.

Oracle Edge Analytics has been migrated from Equinox 3.6 to Equinox 3.8.1 in this release.

Oracle Edge Analytics includes a subset of the functionality available in Oracle Event Processing. It also includes functionality specific to embedded environments. For a list of features both included and omitted, see [Features](#).

1.1.1 Use Cases

The use cases described in this section illustrate specific uses for Oracle Edge Analytics.

Temperature Analysis

A company provides smart home services, in part by monitoring temperature events from residential thermostats. The company wants finer control over event analysis and to reduce network traffic.

The company receives temperature events over the Internet from thermostats in many locations. Data from the thermostats is used to identify patterns and possible alert conditions.

Within an Oracle Edge Analytics application embedded in the thermostat devices, Oracle CQL queries aggregate the event data and perform a threshold analysis before the events are sent over the Internet. As a result, the events received have already been identified as worth attention.

Server Room Monitoring

A company offering data management devices and services needs to improve its data center coordination and energy management to reduce the total cost of ownership. The company needs finer-grained, more detailed sensor and data center reporting.

The company receives energy usage sensor data from disparate resources. Data from each sensor must be analyzed for its local relevance, then must be aggregated with data from other sensors to identify patterns that can be used to improve efficiency.

Separate Oracle Event Processing applications provide a two-tiered approach. In one tier, sensors at the very edge of the network represent event sources that send data to gateway devices. These devices run Java Standard Edition Embedded and Oracle Edge Analytics.

Oracle Edge Analytics applications running on the devices use Oracle Continuous Query Language (Oracle CQL) to query and filter events generated by the sensors. Only event data that meets the filtering criteria is sent to back-end servers in the datacenter where Oracle Event Processing applications run. The Oracle Event Processing applications on the servers send alerts when needed and aggregate and correlate data from across the system to identify consistency issues and to produce data to be used in reports on patterns.

Grid Modernization

A company has a requirement to monitor field networks to prevent power grid restarts and save millions of dollars in costs. Communication needs to be established over IP using a light weight framework.

Using Oracle Edge Analytics, deployed EDGE devices establish a node-to-node communication and a *master* EDGE device sends voltage information, power quality, health status, to a centralized management system. This solution dramatically reduces operational costs.

1.1.2 Features

This section lists the features that are supported and those features that are not supported by Oracle Edge Analytics.

Oracle Edge Analytics includes a subset of the features provided in Oracle Event Processing. This subset supports the particular application needs of event processing in an embedded environment.

Note:

There are no tool-related warnings related to using features in an embedded application that are not supported by that tool. When you use Oracle JDeveloper to develop Oracle Edge Analytics applications, use only supported features.

Table 1-1 Features Included in Oracle Edge Analytics

Feature	Notes
Programming model	As with traditional Oracle Event Processing, you develop embedded applications as event processing networks (EPNs). Some features of the programming model are <i>not</i> supported, including features that support tuning for high availability and scalability.
Oracle (CQL)	Oracle CQL is supported, with the exception of functionality provided by the data cartridges listed in Table 1-2 .
Java data cartridge	The Java data cartridge enhances Oracle CQL with the ability to invoke Java code from Oracle CQL code.
Data source access	Even though the JDBC drivers are removed from the embedded profile, using data sources to connect to database through JDBC is supported in this release. You can configure a data source in the standard way using the <code>config.xml</code> file. The JDBC Driver for Java DB (Derby) embedded and JDBC Driver for Oracle Database are pre-packaged with Oracle Edge Analytics 12c release. For more information about database access, see Application and Development Framework .
RESTful web services	You can implement RESTful web services through the Jersey JAX-RS support included with Java Embedded Suite. For more information, see REST Web Services Support .
Local cache	You can use the included local caching service. Coherence caching is not supported in this release. For more about caching support, see Application and Development Framework .
Security	Including authentication and SSL and utilities such as <code>policygen</code> , <code>cssconfig</code> , and <code>encryptMSAConfig</code> .
Jetty (HTTPS) service	For more information on Jetty in Oracle Edge Analytics, see Application and Development Framework .

Table 1-1 (Cont.) Features Included in Oracle Edge Analytics

Feature	Notes
Configuration	You can use the Configuration Wizard silent mode to create and configure domains. For more information, see Creating Domains and Configuring the Server .
Deployer	You can deploy applications from the command line with the Deployer tool. For more information on Deployer support in Oracle Edge Analytics, see Application Deployment .
Administration tool	For more information on using <code>wlevs.Admin</code> , see Application Administration .
Management through JMX	This release supports management through Java Management Extensions (JMX). For more about JMX in Oracle Edge Analytics, see Application Administration .
Logging	Oracle Event Processing servers save information to log files for viewing.

[Table 1-2](#) lists the Oracle Event Processing features that are not supported in Oracle Edge Analytics. If you attempt to deploy an application that uses these features, you can receive a deployment or runtime error or failure.

Table 1-2 Features Not Included in Oracle Edge Analytics

Feature	Notes
Clustering	Embedded applications cannot be clustered.
Coherence caching	Caching through the included local cache service is supported.
Oracle Event Processing Visualizer	Oracle Event Processing Visualizer is not available as a user interface to manage applications. You can manage applications with the command line tools, including the Deployer and the Administration tool.
HTTP publish-subscribe server and adapter	This release does not include the HTTP publish-subscribe server.
JMS adapter and WebLogic Server JMS client API	This release does not include support for JMS.
CSV file adapter	This release does not include an adapter for using a CSV file as test event data.
Load generator	This release does not include the tool for generating event data.
Event recording and playback	This release does not include the ability to record event activity and play it back for testing and debugging.

Table 1-2 (Cont.) Features Not Included in Oracle Edge Analytics

Feature	Notes
JDBC drivers and Berkeley DB	You can configure a data source in the standard way with the <code>config.xml</code> file. This release supports access to Java DB and Oracle DB through the drivers included with Oracle Edge Analytics 12c release.
Oracle CQL Cartridges (except Java)	Of the Oracle CQL cartridges, which provide support for Oracle CQL enhancements, only the Java cartridge is supported. In particular, the spatial and JDBC cartridges are <i>not</i> supported.
Java Persistence API (JPA) for object-relational mapping	This release does not include support for JPA.
Java Architecture for XML Binding (JAXB)	This release does not include support for JAXB.
SOAP web services	This release does not include support for SOAP web services.
Some security utilities	The <code>passgen</code> , <code>secgen</code> , and <code>GrabCert</code> utilities are not supported in this release.
Monitoring	This release does not include support for monitoring.
Event inspect and trace	This release does not include features to view events flowing from EPN stages and inject events into EPN stages.
Oracle Stream Explorer visual experience	A methodology for using Oracle Stream Explorer Tooling is provided.

Note:

You can use some unsupported features early in development, such as the load generator and CSV adapter, if you remove them before you deploy the application into an embedded environment.

1.1.3 Platform Support

Oracle Edge Analytics runs on any platform that can run Java Virtual Machine Server of Java Standard Edition Embedded 1.7.* and 1.8.*. However, Server Java Virtual Machine is supported only for the following processors:

- ARM v7 Hard Float
- ARM v7 Soft Float
- i586.

Note:

Use Client Java Virtual Machine if the processors you use are other than the ones listed above.

The following table is the matrix for supported platforms:

Table 1-3 Supported Platforms

Platform	Operating System (OS)
Linux	x86-64, oel6/oel5
Raspberry Pi	32-bit Raspbian GNU/Linux 7 \n \l

Note:

When you try to start Oracle Event Processing Server default domain using JDK Version 8 Update 6 for ARM on Raspberry Pi devices, the following error appears:

Error occurred during initialization of VM Server. VM is only supported on ARMv7+ VFP.

To overcome this problem, replace `-server` with `-client` within the JVM Arguments in the `startwlevs.sh` file.

1.1.4 Supported Java Versions

The following table summarizes the supported Java Platforms for Oracle Edge Analytics and Oracle Edge Analytics with Eurotech Bundles.

Table 1-4 Supported Java Versions

Java Edition	x86-64, Linux	Raspberry PI Model B+ (armv6, 32-bit Raspbian GNU/Linux 7 \n \l)
Java Standard Edition 7	JDK 1.7.0.55	not supported
Java Standard Edition 8	JDK 1.8.0.45	JDK 1.8.0_06 (supported on client JVM only)
Java Standard Edition Embedded 7	Java Standard Edition Embedded Runtime Environment 1.7.0_75 (supported on client JVM only)	Java Standard Edition Embedded Runtime Environment 1.7.0_75 (supported on client JVM only)
Java Standard Edition Embedded 8	Java(TM) SE Embedded Runtime Environment 1.8.0_33	Java Standard Edition Embedded Runtime Environment 1.8.0_33 (supported on client JVM only)

1.2 PreLoading and PostLoading

PreLoading and *PostLoading* provide a functionality to load user-defined bundles before and after loading the Oracle Event Processing bundles.

Create the following files for preloading and postloading:

- `oep_domain_pre.xml` in `<domain>/<server>/modules/ext` folder for preloading
- `oep_domain_post.xml` in `<domain>/<server>/modules` folder for postloading.

Note: These files can contain the bundles to be loaded in the format similar to that of the `bundleloader.xml` files. `startlevel` is ignored in `oep_domain_pre.xml` (because it has to be loaded before the Oracle Event Processing bundles).

```
<bundleloader>
  <bundle>
    <startlevel>1</startlevel>
    <location>user_projects/domains/oep_domain/defaultserver/customer/plugins</location>
    <name>org.eclipse.equinox.console_1.0.0.v20120522-1841.jar</name>
  </bundle>
</bundleloader>
```

The `startlevel` indicates the order of starting the bundles. For the bundles having the same `startlevel`, it is loaded as per the order in the XML file. It is a best practice to limit the `startlevel` to a small number say 4 to 6. It can start from 1. The `startlevel` can also be skipped. If it is skipped it will start at the default `startlevel`. This mechanism calculates the offset to start the bundles relative to the `startlevel` of the Oracle Event Processing bundles. You do not need to worry about offsets. The `location` specifies the path from where you can load the bundles. This path is relative to the Oracle Event Processing install directory. The `name` is the name of the bundle to be loaded.

`oep_domain_pre.xml` in `<domain>/<server>/modules/ext` folder

The bundles in this file will load before the Oracle Event Processing bundles and the `startlevel` is ignored. Define bundle `derby1.jar` in the `oep_domain_pre.xml` in `<domain>/<server>/modules/ext` folder, then the Oracle Event Processing will start `derby1.jar` at `startlevel 3`, but before the Oracle Event Processing bundles.

```
<bundle>
  <location>customer/plugin</location>
  <name>derby1.jar</name>
</bundle>

163|Active      | 3|file:/scratch/<username>/Oracle/embedded-release-12.1.3.0-
SNAPSHOT-oep/oep/..../customer/plugin/derby1.jar
```

oep_domain.post.xml in <domain>/<server>/modules folder

The bundles in this file will load after the Oracle Event Processing bundles and the startlevel indicates the order of starting the bundles. There is no validation for startlevel values, whether it is a positive value or a negative value. You need to validate the xml file, the code does not handle any validations.

```
<?xml version="1.0" ?>
<bundleloader>
<bundle>
<startlevel>4</startlevel>
<location>customer/plugin</location>
<name>derby1.jar</name>
</bundle>
</bundleloader>

200|Active | 8|file:/scratch/<username>/Oracle/embedded-release-12.1.3.0-SNAPSHOT-
oep/oep/./customer/plugin/derby1.jar
```

1.3 Application and Development Framework

Oracle Edge Analytics supports Java Embedded Suite (JES) technologies such as Java DB for database access, Jersey for RESTful web services support, and the Java Standard Edition Embedded JRE. You can use the jconsole to monitor Oracle Edge Analytics remotely.

For JES documentation, see <http://www.oracle.com/technetwork/java/embedded/resources/java-embedded-suite/index.html>.

The following describes technologies that are part of the framework that supports Oracle Edge Analytics.

Database Access

Oracle Edge Analytics supports the Java DB database engine (based on the Derby project). You can use Java DB in either client mode or embedded server mode. You enable access to a Java DB database by configuring the data source in the server configuration file. Oracle Edge Analytics 12c only includes Derby Embedded JDBC driver and Oracle DB JDBC driver. For other JDBC drivers, you need to copy the driver to the correct location.

Table 1-5 lists Derby drivers and how you can get them:

Table 1-5 Location of Derby Drivers to Support Java DB Access

Driver	Where to Find It
Embedded Java DB driver	The embedded driver, <code>derby.jar</code> , is pre-packaged with Oracle Edge Analytics.
Java DB client driver (used for client access to a remote Java DB instance)	The client driver, <code>derbyclient.jar</code> , is not pre-packaged with Oracle Edge Analytics 12c. You can download the correct version at the following URL: http://db.apache.org/derby/releases/release-10.8.2.2.html

Once you have the driver JAR file, copy it to the following location in the Oracle Edge Analytics server file system:

`DOMAIN_HOME\SERVER_HOME\modules\ext`

With the driver JAR in the correct location, you can configure data sources through a `data-source` element that you add to the server `config.xml` file.

Web Server

Oracle Edge Analytics includes the Jetty web server and supports the Java Servlet API. The standard `org.osgi.service.http.HttpService` interface is also supported, allowing servlets to be dynamically registered.

Note: Do not change the default Jetty Web Server name in the `config.xml` file. If you change the name, the Oracle Event Processing server crashes.

Cache Support

As in Oracle Event Processing, Oracle Edge Analytics includes simple caching support. You can use the local cache to write applications that access cached data for faster processing. Note that Oracle Coherence is not supported.

For more information about the local cache, see *Developing Applications for Oracle Event Processing*

REST Web Services Support

You can implement web services to expose aspects of your Oracle Edge Analytics applications. Oracle Edge Analytics enables you to implement RESTful web services based on the JAX-RS standard with the Jersey libraries pre-packaged with Oracle Edge Analytics. Jersey is the reference implementation of the Java API for RESTful Web Services (JAX-RS) standard.

For more on enabling REST support in your Oracle Edge Analytics application, see [REST Web Services Support](#).

For more information about Jersey, see the Jersey User Guide at <http://jersey.java.net/nonav/documentation/latest/user-guide.html>.

1.4 Creating Domains and Configuring the Server

Use the Configuration Wizard to create a new domain on which to deploy Oracle Edge Analytics. Oracle Edge Analytics supports the Configuration Wizard in silent mode, a non-interactive way to create and configure a domain.

The Configuration Wizard creates a single default server in the domain. All of the server-related files are located in a subdirectory of the domain directory that is named the same as the server.

Note:

The public-key certificate must include the full host name as its `cn` name and the certificate alias name should be `evsidentity`.

1.4.1 Run the Configuration Wizard in Silent Mode

Perform the following steps to run the configuration wizard in silent mode.

1. Create a `silent.xml` file that defines the domain configuration settings.

Incorrect entries in the `silent.xml` file can cause failures. To determine the cause of a failure, create a log file when you launch the Configuration Wizard.

2. Open a command window and change to the `/<Oracle_Home>/oep/common/bin` directory.

3. Invoke the `config.sh` shell command in silent mode:

```
sh config.sh -mode=silent -silent_xml=path_to_xml_file
```

where `path_to_xml_file` is the full path name of the `silent.xml` template file you created.

4. To create an execution log, use the `-log=full_path_to_log_file` option:

```
sh config.sh -mode=silent -silent_xml=path_to_xml_file -log=/home/logs/create_domain.log
```

Upon successful completion, the command returns messages similar to the following:

```
/home/testuser/java/jes7.0/jre/bin is added to path
<Tue Feb 26 01:41:23 PST 2013> <Info> <BootBundle> <BEA-1004030> <An encryption
key file at location /home/testuser/test_domain/test_server/.aesinternal.dat has
been generated>
testuser@emb-sca-ti-xm-2:~/Oracle/Middleware0220/ocep_11.1/common/bin$
```

If the Configuration wizard does not complete successfully, check the log file for more information. The shell command exit code can help you learn more about the outcome of silent execution.

1.4.2 Create a `silent.xml` File

This section provides the steps to create a `silent.xml` file.

1. Create an empty file called `silent.xml` on the computer on which you want to run the Configuration Wizard in silent mode.
2. Copy the contents of the following sample XML file into your own `silent.xml` file.
3. In the `silent.xml` file you just created, edit the values for the keywords shown in the table below to reflect your configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<bea-installer xmnlls="http://www.bea.com/platend/wlevs/config/
silent">
  <data-value name="CONFIGURATION_OPTION" value="createDomain" />
  <data-value name="USER-NAME" value="wlevs" />
  <data-value name="PASSWORD" value="wlevs" />
  <data-value name="SERVER_NAME" value="my_server" />
  <data-value name="DOMAIN_NAME" value="mydomain" />
  <data-value name="DOMAIN_LOCATION" value="/home/mydomains" />
  <data-value name="NETIO_PORT" value="9002" />
```

```

<data-value name="KEYSTORE_PASSWORD" value="welcome1" />
<data-value name="DB_URL" value="jdbc:oracle:thin:@hostname.com:1521:xe" />
<data-value name="DB_USERNAME" value="wlevs" />
<data-value name="DB_PASSWORD" value="wlevs" />
</input-fields>
</bea-installer>

```

4. Save the file in the directory of your choice.

Table 1-6 Data-value names for silent.xml file

For this data-value name...	Enter the following value...
CONFIGURATION_OPTION	Specifies whether you want to create a new domain with a default server or update a server in an existing domain. Valid values are <code>createDomain</code> or <code>updateDomain</code> . Default value is <code>createDomain</code> .
USERNAME	The user name of the administrator of the created or updated server in the domain.
PASSWORD	The password of the administrator of the created or updated server in the domain.
SERVER_NAME	The name of the new server in this domain. This name is also used for the name of the directory that contains the server files.
DOMAIN_NAME	The name of the domain.
DOMAIN_LOCATION	The full name of the directory that contains the domain.
NETIO_PORT	The port number to which the server instance itself listens.
KEYSTORE_PASSWORD	The password for the identity key store.
PRIVATEKEY_PASSWORD	The password for the certificate private key. The default value of this option is the value of the <code>KEYSTORE_PASSWORD</code> .
DB_URL	The URL to connect to a database with JDBC. This option is used to configure the data source. The database configuration parameters are optional; if you do not specify them, then no data source is configured for the server. The configuration wizard supports configuring a data source using the Oracle database only. To use another database technology, please edit the configuration manually.

Table 1-6 (Cont.) Data-value names for silent.xml file

For this data-value name...	Enter the following value...
DB_USERNAME	<p>The name of the user that connects to the database through the data source.</p> <p>The database configuration parameters are optional. If you do not specify them, then no data source is configured for the server.</p>
DB_PASSWORD	<p>The password of the user that connects to the database via the data source.</p> <p>The database configuration parameters are optional. If you do not specify them, then no data source is configured for the server.</p>

When run in silent mode, the Configuration Wizard generates exit codes that indicate the success or failure of domain creation and configuration. These exit codes are shown in the following table:

Table 1-7 Configuration Wizard Silent Mode Exit Codes

Code	Description
0	Configuration Wizard execution completed successfully.
-1	Configuration Wizard execution failed due to a fatal error
-2	Configuration Wizard execution failed due to an internal XML parsing error

1.5 Applications Development

You can develop Oracle Edge Analytics applications with Oracle JDeveloper. See *Getting Started with Oracle Event Processing* for information about Oracle JDeveloper.

When you develop embedded applications, you must limit the set of features you use to those that are supported on Oracle Edge Analytics.

Note:

Oracle JDeveloper currently includes no support for ensuring that the embedded applications you build will deploy and run in an embedded environment. For example, Oracle JDeveloper does not validate that your embedded application includes features that are not supported.

If you create applications using features unsupported by Oracle Edge Analytics (whether or not you use Oracle JDeveloper), you might see deployment errors or runtime failures.

For a list of Oracle Event Processing features included in (and excluded from) Oracle Edge Analytics, see [Features](#).

1.5.1 Example Embedded Application

Oracle Edge Analytics includes the Smart Appliance example application. The application listens for device events through a socket connection. The device events represent device information and the values associated with it. The threshold values for each device is stored in a JavaDB database. As the application processes device events, if an event value does not fall within specified threshold limits, then the application generates the alerts.

For more information about the sample, including how to run it, see the `readme.txt` file included with it.

After you install Oracle Edge Analytics, you can find the Smart Appliance application at the following file system location:

```
/<Oracle_Home>/oep/examples/source/smartzappliance
```

1.5.2 REST Web Services Support

You can expose RESTful web services from your application by using Jersey, the JAX-RS implementation pre-packaged with Oracle Edge Analytics. Once you have REST support enabled, you can annotate methods of your Java code so that they execute when REST client calls are made. The Smart Appliance sample application is an example of a REST-enabled application.

For more information about using Jersey and JAX-RS, see the Jersey User Guide at <http://jersey.java.net/nonav/documentation/latest/user-guide.html>.

Enable Support for REST Web Services

To enable REST support, ensure that the Jersey libraries are in the application class path so that you can use them in classes that implement methods as REST resources. Also configure your RESTful classes with the OSGi service so that they are available to external clients.

The following REST web services are supported:

- Inbound REST with JSON, XML, and CSV media types
- Outbound REST with JSON and XML media types.

To enable REST web services support:

1. Ensure that the Jersey libraries are in your application class path.

For example, put the JAR files in the `modules/ext` directory of your Oracle Edge Analytics server domain. On 12c Oracle Edge Analytics, jersey jar files are pre-packaged.

2. In your application assembly file, include an OSGi HTTP Service reference and inject it into the bean that represents your RESTful class, as follows:

```
<osgi:reference id="httpServiceRef"
interface="org.osgi.service.http.HttpService" />
<!-- Bean declaration with properties for the RESTful service root context
and OSGi HTTP service.  -->
<bean id="bean" class="com.oracle.cep.example.smartzappliance.sink.AlertsService">
  <property name="rootContext" value="/alerts" />
  <property name="httpService" ref="httpServiceRef" />
</bean>
```

3. In your bean class code, provide a setter method for the HTTP Service, such as `setHttpService(HttpServletRequest httpService)`.
4. Register a REST servlet with the OSGi HTTP service as follows:
 - a. Create a Jersey `ServletContainer` instance with the appropriate initialization parameters.
 - b. In the initialization parameters, set the `javax.ws.rs.Application` property as the implemented JAX-RS Application class name. The `Application` class is the one that loads the resource classes.

The following example shows the `setHttpService(HttpServletRequest httpService)` setter method, and the OSGi service registration.

```
public void setHttpService(HttpServletRequest httpService)
    throws Exception {
    Hashtable<String, String> initParams = new Hashtable<String, String>();
    initParams.put("javax.ws.rs.Application",
        WebApplication.class.getName());
    Servlet jerseyServlet = new ServletContainer();
    httpService.registerServlet(rootContext, jerseyServlet, initParams, null);
}
```

1.6 Application Deployment

You deploy Oracle Edge Analytics applications with the Deployer command-line tool included in the installation. You can deploy over both SSL and or non-SSL connections.

The Deployer tool is nearly identical to the Deployer tool provided in Oracle Event Processing, but cluster and group deployments are not supported in Oracle Edge Analytics.

Note:

Oracle Event Processing Visualizer is not included with Oracle Edge Analytics, so deployment through a graphical user interface is not supported.

To deploy an application to a running Oracle Edge Analytics server, use a command similar to the following:

```
java -jar wlevsdeploy.jar -url http://hostname:9002/wlevsdeployer -user oepadmin -
password welcome1 -install myoepapp.jar
```

The following examples illustrate additional deployment scenarios.

Deploy to a Local Server

```
java -jar wlevsdeploy.jar -user oepadmin -password welcome1 -url http://localhost:
9002/wlevsdeployer -install application
```

Deploy Over a Non-SSL Connection

```
java -jar wlevsdeploy.jar -user oepadmin -password welcome1 -url http://<host>:9002/
wlevsdeployer -install application
```

Deploy Over SSL

```
java -Djavax.net.ssl.trustStore=d:\downloads\evsidentity.jks -  
Djavax.net.ssl.trustStorePassword=welcome1 -jar wlevsdeploy.jar -user oepadmin -  
password welcome1 -url https://<host>:9003/wlevsdeployer -install application
```

1.7 Application Administration

You can administer Oracle Edge Analytics applications with the `wlevs.Admin` command-line tool and Java Management Extensions (JMX). Whether you use JMX or `wlevs.Admin`, Oracle Edge Analytics supports administration for all but those features omitted from this release. For more information about administrative features, see *Administering Oracle Event Processing*.

For a list of omitted features, see [Features](#).

Note:

Oracle Event Processing Visualizer is not included with Oracle Edge Analytics, so administration through a graphical user interface is not supported.

Administration with JMX

Oracle Edge Analytics supports remote JMX connections through MSA RMI as in Oracle Event Processing. Most configuration and runtime MBeans are supported, with the exception of those specifically related to omitted features. For example the `HttpPubSubAdapterMBean`, `JMSAdapterMBean`, and `Monitoring` related MBeans are not supported because the underlying features are not supported.

Note:

The Java Standard Edition Embedded JRE does not include support for the JMX local attach API, so local (non-RMI) JMX connections aren't supported.

Administration with `wlevs.Admin`

The `Admin` command line utility is supported and includes most of the functionality present in Oracle Event Processing. Unsupported functionality is specifically related to unsupported features, including commands related to monitoring or record and play back.

For example, you can add an Oracle CQL rule locally with the `Admin` utility `ADDRULE` command, as shown in the following example:

```
prompt> java wlevs.Admin  
-url service:jmx:msarmi://localhost:9002/jndi/jmxconnector  
-username wlevs -password wlevs  
ADDRULE -application helloworld -processor helloworldProcessor  
-query myquery "SELECT * FROM Withdrawal [Rows 5]"
```

Note:

This release does not support starting the server with the `-disablesecurity` flag.

1.7.1 The `policygen` Command-Line Utility

Use the `policygen` command-line utility to generate an extensible access control markup language (XACML) file.

The `policygen` utility is located at `/<Oracle_Home>/oep/bin/`. Use the following syntax to form the command:

```
policygen.sh [-h] [-s] [-l|-x] entitlement_file xacml_output_file
```

Table 1-8 policygen Arguments

Option	Description	Defaults
<code>-s</code>	Specifies that standard XACML policy is generated in the output file; otherwise, entitlement XACML policy will be generated.	
<code>-l</code>	Specifies that an XACML LDIFT file should be generated.	
<code>-x</code>	Specifies that an XACML policy file should be generated.	
<code>-h</code>	Displays usage help.	
<code>entitlement_file</code>	Specifies the location of input entitlement XML file.	
<code>xacml_output_file</code>	Specifies the location of output XACML file.	

For example:

```
prompt> policygen.sh ~/security/policygen/AuthorizerInit.xml
XACMLAuthorizerInit.ldift
```

1.8 Supported Platform and Resource Configurations

Oracle Edge Analytics supports the following platform configurations:

- Java Standard Edition Embedded 7 and Java Standard Edition Embedded 8
- Operating System: Linux
- Hardware: x86 or ARM

Oracle Edge Analytics supports smaller disk and heap footprint requirements. For a new installation, it requires 83 MB of disk space. With the server started and applications deployed, additional disk space is needed per application.

Installation

This chapter describes how to install Oracle Edge Analytics.

This chapter includes the following sections:

- [Installation Overview](#)
- [Install from the Command Line](#)
- [Post-Installation Steps](#).

2.1 Installation Overview

Oracle Edge Analytics does not include an installation program. To install it, you extract the contents of its distribution file to the location where you want to install it.

Oracle Edge Analytics is designed for use with the Java Standard Edition Embedded. The installation instructions provided here assume that your installation of Oracle Edge Analytics will use the JRE provided with Java Standard Edition Embedded.

To install Oracle Edge Analytics:

1. Download and install Oracle Edge Analytics.
2. Set `JAVA_HOME` to use the Java Standard Edition Embedded JRE.

For more detailed step-by-step instructions, including download location, see [Install from the Command Line](#).

2.1.1 Before You Install

Oracle Edge Analytics is designed to work with Java Standard Edition Embedded. Because of this, Java Standard Edition Embedded is a prerequisite for using Oracle Edge Analytics.

The installation instructions include notes about Java Standard Edition Embedded installation.

For additional system requirements, see [Supported Platform and Resource Configurations](#).

2.1.2 Directory Structure and Concepts

The Oracle Edge Analytics distribution file includes the event server, a default domain, and a sample application.

When you extract the Oracle Edge Analytics distribution file, the directory structure includes the following noteworthy items:

Table 2-1 Oracle Edge Analytics Directory Structure

Table 2-1 (Cont.) Oracle Edge Analytics Directory Structure

Path	Description
OEPEMBEDDED_HOME/oep/modules	Libraries comprising product features.
OEPEMBEDDED_HOME/oep/bin	Configuration and deployment tools.
OEPEMBEDDED_HOME/oep/common	Tools and examples for domain configuration.
OEPEMBEDDED_HOME/oep/examples	Sample application code.
OEPEMBEDDED_HOME/oep/utils	Database and security utilities.
OEPEMBEDDED_HOME/oep/xsd	XML schemas that define XML configuration files.
OEPEMBEDDED_HOME/user_projects	Default domain for user projects.

2.2 Install from the Command Line

You install Oracle Edge Analytics by extracting the contents of the installation *.zip file to the directory of your choosing.

Note:

There is no graphical installer in this release.

To install Oracle Edge Analytics:

To get started, download an ejDK bundle suitable for your target platform and follow instructions to create a JRE that suits your application's needs for Java Standard Edition Embedded 8. This change does not affect JRE downloads for Java SE Embedded 7 Update releases. For instructions, see <http://docs.oracle.com/javase/8/embedded/develop-apps-platforms/jrecreate.htm#JEMAG270>.

1. Log in to the device on which you want to install Oracle Edge Analytics. Be sure you log in as the user that will be the main administrator of the installation.
2. Download the Oracle Edge Analytics distribution ZIP file ofm_oea_generic_12.1.3_disk1_1of1.zip from Oracle Technology Network at: <http://www.oracle.com/technetwork/middleware/complex-event-processing/downloads/index.html>.
3. Extract the file to the installation location using the following command:

```
unzip ofm_oea_generic_12.1.3_disk1_1of1.zip
```

4. Check the Java Standard Edition Embedded version you need from Oracle web site at <http://www.oracle.com/technetwork/java/embedded/embedded-se/overview/index.html>.
5. Download the Java Standard Edition Embedded distribution ZIP file.
6. Extract the file to the Java home location. For example, use a command such as the following:

```
tar -zxf ejre-7u75-fcs-b13-linux-arm-vfp-hflt-client_headless-18_dec_2014.tar.gz
-C /home/pi
```
7. Ensure that you have a \$JAVA_HOME system variable and set it to the path where you unzipped the Java Standard Edition Embedded.

```
export JAVA_HOME=/home/pi/ejre1.7.0_75
```

2.3 Post-Installation Steps

For information about configuring and domain creation, see the following sections:

- [Creating Domains and Configuring the Server](#)
- [Application and Development Framework](#).

