Oracle® Fusion Middleware

Installing and Configuring Oracle Stream Analytics

12c (12.2.1.2)

E77599-02

November 2016

This document describes how to install and configure Oracle Stream Analytics (OSA).

About the Oracle Stream Analytics Installation

This document explains how to install and configure a new Oracle Stream Analytics 12*c* (12.2.1.2) Oracle home.

After you install Oracle Stream Analytics (OSA), you can configure a standalone OSA domain. Note that an OSA standalone-server domain does not require Oracle WebLogic Server. For more information, see Standalone-Server Domains in *Oracle Fusion Middleware Administering Oracle Stream Explorer*. Optionally, you can integrate Apache Spark with your OSA installation, as described in Installing and Configuring Apache Spark.

If you are using a previous version of Oracle Event Processing (the prior name for OSA), note there is no upgrade to the OSA 12*c* (12.2.1.2) runtime software. If you created an Oracle Event Processing 11*g* standalone-server domain, then you must install a new OSA 12*c* (12.2.1.2) Oracle home and configure a new OSA 12*c* (12.2.1.2) standalone-server domain.

Roadmap for Verifying Your System Environment

Before you begin the installation and configuration process, you must verify your system environment.

The following table identifies important tasks and checks to ensure that your environment is properly prepared for installing and configuring OSA.

Table 1-1 Roadmap for Verifying Your System Environment

Roadmap for Verifying Your System Environment

Task	Description	Documentation
Verify certification and system requirements.	Verify that your operating system is certified and properly configured for installation and configuration.	See Verifying Certification, System Requirements, and Interoperability in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.
Identify a proper installation user.	Verify that the installation user has the proper permissions to install and configure the software.	See Selecting an Installation User in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.
Select the installation and configuration directories on your system.	Verify that you are able to create the necessary directories for installation and configuration, according to the recommended directory structure.	See About the Directories for Installation and Configuration in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.
Install a certified JDK.	The installation program for the distribution requires a certified JDK present on your system.	See About JDK Requirements for an Oracle Fusion Middleware Installation in *Oracle Fusion Middleware Planning an Installation of Oracle Fusion Middleware*.

Obtaining the Product Distribution

The distribution for Oracle Stream Analytics (OSA) is available on the Oracle Technology Network (OTN).

To obtain OSA:

1. Go to the OSA Installer download page on OTN at http://www.oracle.com/technetwork/middleware/complex-event-processing/download/index.html
2. Accept the license agreement, then select **+Recommended Install Process**, then the **download** link for the Oracle Stream Analytics Installer (`fmw_12.2.1.2.0_osa_Disk1_1of1.zip`).
3. Extract the contents of this `.zip` file onto your system. One of the files extracted will be `fmw_12.2.1.2.0_osa_generic.jar`, which runs the product installer and installs the software onto your system (see Installing Oracle Stream Analytics).

Installing Oracle Stream Analytics

This section describes how to install the OSA software and create the Oracle home directory.

- Starting the Installation Program

- Navigating the Installation Screens
- Verifying the Installation

Starting the Installation Program

Before running the installation program, you must verify the JDK and prerequisite software is installed.

To start the installation program:

1. Sign in to the host system.
2. If you have not already done so, verify that a certified JDK is installed on your system: enter `java -version` on the command line. For 12*c* (12.2.1.2), the certified JDK is 1.8.0_101 and later.

   For more information, see About JDK Requirements for an Oracle Fusion Middleware Installation.
3. Verify that you have installed all prerequisite software, such as Oracle Fusion Middleware Infrastructure.
4. Go to the directory where you downloaded the installation program.
5. Start the installation program by running the `java` executable from the JDK directory. For example:

   - (UNIX) `/home/Oracle/Java/jdk1.8.0_101/bin/java –jar fmw_12.2.1.2.0_osa_generic.jar`
   - (Windows) `C:\home\Oracle\Java\jdk1.8.0_101\bin\java -jar fmw_12.2.1.2.0_osa_generic.jar`

Note:

You can also start the installer in silent mode using a saved response file instead of launching the installer screens. For more about silent or command line installation, see Using the Oracle Universal Installer in Silent Mode in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

The installer shows a series of screens where you verify or enter information.

The following table lists the order in which installer screens appear. If you need additional help with an installation screen, click **Help**.

Table 1-2 Oracle Stream Analytics Install Screens

Oracle Stream Analytics Install Screens

Table 1-2 (Cont.) Oracle Stream Analytics Install Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen opens if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location. For more about the central inventory, see About the Oracle Central Inventory in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*. This screen does not appear on Windows operating systems.
Welcome	On this screen, review the information to make sure that you have met all the prerequisites, then click **Next**.
Auto Updates	On this screen, select to skip automatic updates, select patches, or search for the latest software updates, including important security updates, through your My Oracle Support account.
Installation Location	Use this screen to specify your Oracle home directory location. You can click **View** to verify and ensure that you are installing Oracle Stream Analytics in the correct Oracle home.
Installation Type	Select either **Stream Analytics** or **Stream Analytics With Examples**. Toggle the option you prefer and review the items that will be installed in the list below the installation types.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. To view the list of tasks that gets verified, select **View Successful Tasks**. To view log details, select **View Log**. If any prerequisite check fails, then an error message appears at the bottom of the screen. Fix the error and click **Rerun** to try again. To ignore the error or the warning message and continue with the installation, click **Skip** (not recommended).
Installation Summary	Use this screen to verify installation options you selected. If you want to save these options to a response file, click **Save Response File** and enter the response file location and name. The response file collects and stores all the information that you have entered, and enables you to perform a silent installation (from the command line) at a later time. Click **Install** to begin the installation.
Installation Progress	This screen shows the installation progress. When the progress bar reaches 100% complete, click **Finish** to dismiss the installer, or click **Next** to see a summary.
Installation Complete	This screen displays the Installation Location and the Feature Sets that are installed. Review this information and click **Finish** to close the installer.

Verifying the Installation

After you complete the installation, verify it was successful by completing a series of tasks.

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that the installer did not encounter any problems.

By default, the installer writes logs files to the *`Oracle_Inventory_Location`*`/logs` (on UNIX operating systems) or *`Oracle_Inventory_Location`*`\logs` (on Windows operating systems) directory.

For a description of the log files and where to find them, see Installation Log Files in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options you selected during the installation.

For more information about the directory structure after installation, see What Are the Key Oracle Fusion Middleware Directories? in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware*.

Viewing the Contents of the Oracle Home

You can view the contents of the Oracle home using the `viewInventory` script.

For more information, see Viewing the Contents of an Oracle Home in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Configuring the Oracle Stream Analytics Domain

An OSA installation does not include Oracle Fusion Middleware Infrastructure, so only standalone-server domains may be created for OSA. You can learn more about standalone-server domains by reading Standalone-Server Domains in *Oracle Fusion Middleware Administering Oracle Stream Explorer*.

For instructions on configuring a standalone domain, see Create a Standalone-Server Domain in *Oracle Fusion Middleware Administering Oracle Stream Analytics*.This guide also contains other administrative tasks for OSA, including updating a domain and starting and stopping the servers in the domain.

Installing and Configuring Apache Spark

Apache Spark (Spark) is an open source big data processing framework built around speed, ease of use, and sophisticated analytics.

You can use Spark with Oracle's Continuous Query Language (CQL) to scale complex event processing applications on commodity clusters. This means that Spark allows you to process larger volumes of streaming data at a lower cost.

Note: OSA supports only cluster mode Spark deployments.

The installation steps presented here assume the following:

- OSA is installed in *OSA_HOME*.

  For example, *OSA_HOME* `= /apps/oracle/middleware`.

- The OSA application domain has been created, named *OSA_DOMAIN*.

  For example, *OSA_DOMAIN*`=`*OSA_HOME*`/user_projects/domains/osa/defaultserver`.

  Note that there is no step specific to Spark installation when you create the OSA application domain.

- A Spark cluster is set up, using one of the cluster types supported by OSA (such as Spark standalone or Hadoop YARN). For links to the supported Spark version download and documentation, see Prerequisites for Apache Spark Integration.

When you are ready, use the information in the following sections to integrate Spark with OSA:

- Prerequisites for Apache Spark Integration
- Installing Apache Spark
- Installing the OSA-Spark Integration Component
- Installing Kafka
- Configuring the OSA Domain for Spark
- Configuring the osa.properties File

Prerequisites for Apache Spark Integration

Using Spark with OSA requires installing and configuring third party components.

The following table provides information about the components that are required in a Spark cluster environment to use Spark with OSA. This information is referenced from the subsequent topics that describe installing and configuring these components.

Component	Version	Links
Apache Kafka	0.8.2.2 for Scala 2.10	Download: http://kafka.apache.org/downloads.html Documentation: http://kafka.apache.org/documentation.html
Hadoop YARN	2.6.X (recommended)	Download: http://hadoop.apache.org/releases.html
Apache Spark	1.5.X - Prebuilt for Hadoop 2.6 (or your Hadoop version) Note: 1.6.X not supported	Download: http://spark.apache.org/downloads.html Documentation: http://spark.apache.org/docs/1.5.1/

Installing Apache Spark

Install the version of Spark that is supported by your version of OSA.

To install Spark:

- Refer to Prerequisites for Apache Spark Integration to install Spark in *SPARK_HOME*.

Notes:

- OSA supports only cluster mode Spark deployments.
- Install the version of Spark that is supported by your version of OSA. For 12*c* (12.2.1.2), see Prerequisites for Apache Spark Integration.
- If you want to use Spark together with Hadoop Distributed File System (HDFS) and YARN, install the Spark version that was compiled for the Hadoop version compatible with your current environment.
- Your Spark distribution must also contain *SPARK_HOME*`/lib/spark-examples-X.X.X-hadoopY.Y.Y.jar`, which is also required by OSA.
- Spark must be installed on each node in your cluster, and also on the node where the OSA domain runs.
- *SPARK_HOME* must have the same value on each node, including the OSA node. It makes sense to mount the same network drive to the same location on each node.

Installing the OSA-Spark Integration Component

The OSA-Spark integration component adds Oracle's Continuous Query Language (CQL) support, along with an OSA-specific runtime environment, to the Spark framework to implement application deployment.

This component is delivered as part of the OSA server installation as a single JAR at *OSA_HOME*`/oep/spark/lib/spark-osa.jar`.

This JAR file must be copied to all worker nodes and also to the OSA node. Ideally, you copy this file into your Spark installation at *SPARK_HOME*`/lib/spark-osa.jar`.

Installing Kafka

Apache Kafka is required by an OSA-Spark environment to send and receive data.

Kafka is the only external system that an OSA-Spark exploration can accept data from or send data to. If you want to chain two explorations (where one is running in Spark, and the other running in either Spark or OSA), the only way to do this is to explicitly create Kafka targets and streams to connect these explorations. Additionally, Kafka is used to push data from an OSA-Spark exploration back to the "live output stream" of the exploration editor. This is configured in the `osa.properties` file using the `osa.kafka` properties (see Kafka Settings).

To install Kafka:

- Refer to the download and documentation links in Prerequisites for Apache Spark Integration.

Note:

- OSA only needs to know the Kafka endpoints, such as broker and zookeeper addresses.
- OSA uses short-lived Kafka topics to communicate with applications in Spark. To allow OSA to explicitly delete these topics, consider adding this line in the server configuration file *KAFKA_HOME*`/config/server.properties` (where *KAFKA_HOME* is the Kafka installation folder) when installing Kafka:

  ```
  delete.topic.enable=true
  ```

Configuring the OSA Domain for Spark

OSA-Spark integration relies on settings in one or more configuration files.

To configure your OSA domain for Spark:

1. Stop the OSA server (OSA domain).
2. Create the Spark configuration folder in your domain as: *OSA_DOMAIN*`/config/spark`.
3. Create the OSA configuration file in the Spark configuration folder as: *OSA_DOMAIN*`/config/spark/osa.properties`.
4. Edit the OSA configuration file according your environment (see Configuring the osa.properties File).
5. Create any additional configuration files required by your environment in the Spark configuration folder. This may include files specific to Hadoop Distributed File System (HDFS) and YARN, such as `yarn-site.xml`, `core-site.xml`, `hdfs-site.xml`, and more. For details, refer to the documentation for your cluster.

   These files are required by OSA to deploy files to your cluster. OSA needs client-side settings only (such as name node URL, resource manager URL, timeouts, credential settings, and so on). *Do not copy the full server-side YARN/Hadoop configuration.*
6. Edit the content of additional configuration files according your environment.
7. Start your OSA domain.
8. Access your OSA applications at the following URLs:

   - OSA for Spark: `http://`*host*`:`*port*`/sxspark`
   - OSA: `http://`*host*`:`*port*`/sx`

   If you accept the default installation parameters, the URLs are:

   - OSA for Spark: `http://localhost:9002/sxspark`
   - OSA: `http://localhost:9002/sx`

Note: Whenever you deploy an OSA streaming application, the content of the configuration folder is zipped and sent to the Spark cluster. In this way, the same configuration is available for the OSA streaming application while running in the cluster.

Tips:

- Find useful configuration templates for different scenarios in *OSA_HOME*/oep/spark/config-templates. You can start with these files and customize them according to your needs.
- If you want to experiment with different Spark configurations, you can set up multiple configuration folders outside your OSA domain and set up the *OSA_DOMAIN*/config/spark folder as a symlink that points to one of your configuration folders at a time. For example, *OSA_DOMAIN*/config/spark points to your YARN-based OSA configuration in /apps/osa/config/yarn-cluster, and to your Spark Standalone cluster configuration in /apps/osa/config/spark-cluster. In this way, you can easily switch between your configurations.

Configuring the osa.properties File

The osa.properties file is the configuration file for OSA-Spark integration.

Refer to the following sections to set configuration properties:

- General Deployment Settings
- Kafka Settings
- JAR File Settings
- Runtime Settings
- Spark Settings
- Example osa.properties File

General Deployment Settings

General deployment settings in osa.properties define the Spark cluster type and distribution folder.

The following table describes the values for the general deployment settings in osa.properties.

Parameter	Description
`osa.deploy.spark.master`	**Mandatory.** Specifies the Spark cluster type and master URL. You can configure OSA for Spark running on Spark Standalone or YARN cluster: • **Spark Standalone:** The cluster type is identified by `spark://`in the URL. **Important:** you must provide the master's *REST interface* URL. Example: `osa.deploy.spark.master=spark://spark.mycompany.com:6066` • **YARN:** The cluster type is identified by the single word `yarn`. Instead of a master URL, you only enter `yarn`. **Important:** YARN and HDFS-specific client configuration files must be also provided in the configuration folder (`yarn-site.xml`, `core-site.xml`, `hdfs-site.xml`). Example: `osa.deploy.spark.master=yarn`
`osa.deploy.spark.fileshare`	**Mandatory for Spark Standalone. Not used for YARN.** Specifies the shared location that can be used by OSA as a distribution folder for the applications. The folder must be accessible from every node under the same path. This can be an Network File System (NFS) or Hadoop Distributed File System (HDFS) path. If you want to use HDFS, client configuration files (`core-site.xml`, `hdfs-site.xml`) must be provided in the configuration folder. Examples: NFS: `/net/osa/spark-deployments` HDFS: `hdfs://hdfs.mycompany.com/osa/spark-deployments`

Kafka Settings

Kafka settings in `osa.properties` define the Kafka brokers and zookeeper.

OSA exchanges data through Kafka with the streaming application running in Spark. For this reason you have to specify where your Kafka server is installed.

The following table describes the values for the Kafka settings in `osa.properties`.

Parameter	Description
`osa.kafka.brokerlist`	**Mandatory.** Comma-separated list of Kafka brokers in the form *`host`*`:`*`port`*. Example: `kafka.mycompany.com:9092`
`osa.kafka.zookeeper`	**Mandatory.** The Kafka zookeeper in the form *`host`*`:`*`port`*. Example: `zk.mycompany.com:2181`

JAR File Settings

JAR file settings in `osa.properties` define the paths to the Spark assembly and example JAR files.

OSA requires the `spark-assembly` and `spark-examples` packages of your Spark distribution on each worker node and on the OSA node. The path to each JAR must be the same on all nodes. Additionally, the `spark-osa` package must be copied to each node.

The following table describes the values for the JAR file settings in `osa.properties`.

Parameter	Description
`osa.jars.spark-assembly`	**Mandatory.** Path to the `spark-assembly` JAR file. Example: `/apps/spark/spark-1.5.1/lib/spark-assembly-1.5.0-hadoop2.6.0.jar`
`osa.jars.spark-examples`	**Mandatory.** Path to the `spark-examples` JAR file. Example: `/apps/spark/spark-1.5.1/lib/spark-examples-1.5.0-hadoop2.6.0.jar`
`osa.jars.spark-osa`	**Mandatory.** Path to the `spark-osa` JAR file, which you must coppy to each node manually. Example: `/apps/spark/spark-1.5.1/lib/spark-osa.jar`

Runtime Settings

Runtime settings in `osa.properties` tune the deployed OSA streaming applications.

You can tune the deployed OSA streaming applications based on the available resources in your cluster and on the characteristics or nature of the problem that you want to solve with OSA.

The following table describes the values for the runtime settings in `osa.properties`.

Parameter	Default	Description
`osa.runtime.executor.instances`	1	**Optional.** Specifies how many workers will do stream processing in parallel. Note that this number will affect the required executor cores. This parameter value applies to each deployed OSA application.
`osa.runtime.batchDuration`	1000ms	**Optional.** Specifies the batch interval of the streaming data processing.

Spark Settings

Spark settings in `osa.properties` define resource consumption of the deployed Spark application in the cluster.

Note: OSA may override your Spark property settings in `osa.properties` if they are not sufficient to run OSA streaming applications in the Spark cluster.

The following table describes the values for the Spark settings in `osa.properties` that are most typically used in the context of OSA-Spark. Setting values lower than the default values may result in an out of memory error. For more information about Spark property settings, see http://spark.apache.org/docs/1.5.1/configuration.html.

Parameter	Default (minimum)	Description
`spark.executor.memory`	1800m	**Optional.** Amount of memory to use per executor process.
`spark.executor.cores`	1	**Optional.** The number of cores to use on each executor. In standalone mode, setting this parameter allows an application to run multiple executors on the same worker, provided that there are enough cores on that worker. Otherwise, only one executor per application will run on each worker.
`spark.driver.memory`	1500m	**Optional.** Amount of memory to use for the driver process.
`spark.driver.cores`	1	**Optional.** Number of cores to use for the driver process, only in cluster mode.

Example osa.properties File

Review an example configuration in the `osa.properties` file.

```
osa.deploy.spark.master=spark://spark.mycompany.com:6066
osa.deploy.spark.fileshare=/osa-share/spark-deployments

osa.kafka.brokerlist=kbroker1.mycompany.com:9092,kbroker2.mycompany.com:9092
osa.kafka.zookeeper=zk.mycompany.com:2181

osa.jars.spark-assembly=/apps/spark/spark-1.5.1/lib/spark-assembly-1.5.0-
hadoop2.6.0.jar
osa.jars.spark-examples=/apps/spark/spark-1.5.1/lib/spark-examples-1.5.0-
hadoop2.6.0.jar
osa.jars.spark-osa=/apps/spark/spark-1.5.1/lib/spark-osa.jar

osa.runtime.executor.instances=1
osa.runtime.batchDuration=1000ms

spark.executor.cores=2
spark.executor.memory=3g
spark.driver.memory=2500m
```

This example shows:

- Spark runs in a standalone cluster and the OSA applications are distributed through an NFS file share mounted to `/osa-share/spark-deployments` on each node.
- Kafka is installed and configured, and has has two dedicated broker servers and a separate zookeeper server.
- Spark 1.5.1 is installed on each node to `/apps/spark/spark-1.5.1`.
- Runtime parameters specify that there is no parallel processing (single processor instance) and batch interval is set to 1 second.
- Spark-specific parameters specify `spark.executor.cores=2` (defaults to `1`), `spark.executor.memory=3g` (defaults to `1500m` as the minimum), and `spark.driver.memory=2500m` (defaults to `1800m` as the minimum)
- This configuration does not use HDFS or YARN, so the `osa.properties` file is the only configuration file that must be copied to the configuration folder.

Deinstalling the Software

Follow the instructions in this section to start the product deinstaller and remove the software.

If you want to perform a silent (command-line) deinstallation, see Running the Oracle Universal Installer for Silent Deinstallation in *Oracle Fusion Middleware Installing Software with the Oracle Universal Installer*.

Starting the Deinstallation Program

To start the deinstaller:

- **On UNIX**

  On the command line, enter the following commands:

  ```
  cd $ORACLE_HOME/oui/bin
  ./deinstall.sh
  ```

- **On Windows**

  Do one of the following:

  – Use a file manager window to navigate to the *`ORACLE_HOME`*`\oui\bin` directory and double-click on `deinstall.cmd`.

  – Open a command prompt and enter the following commands:

  ```
  cd %ORACLE_HOME%\oui\bin
  deinstall.cmd
  ```

  – From the **Start** menu, select **All Programs**, then **Oracle**, then **OracleHome**, and then **Uninstall Oracle Software**.

Navigating the Deinstallation Screens

The deinstaller shows a series of screens to confirm the deinstallation of the software.

If you need help on screen listed in Table 1-3, click **Help** on the screen.

Table 1-3 Deinstallation Screens and Descriptions

Deinstallation Screens and Descriptions	
Screen	**Description**
Welcome	Introduces you to the product deinstaller.
Deinstallation Summary	Shows the Oracle home directory and its contents that will be deinstalled. Verify that this is the correct directory. If you want to save these options to a response file, click **Save Response File** and enter the response file location and name. You can use response file later during a silent deinstallation. For more on silent or command line deinstallation, see Running the Oracle Universal Installer for Silent Deinstallation in *Installing Software with the Oracle Universal Installer*. Click **Deinstall** to begin removing the software.
Deinstallation Progress	Shows the deinstallation progress.
Deinstallation Complete	Appears when the deinstallation is complete. Review the information on this screen, then click **Finish** to dismiss the deinstaller.

Removing the Oracle Home Directory Manually

After deinstalling the software, you must manually remove your Oracle home directory and any existing subdirectories that the deinstaller did not remove.

For example, if your Oracle home directory is `/home/Oracle/product/ORACLE_HOME` on a UNIX operating system, enter the following commands:

```
cd /home/Oracle/product
rm -rf ORACLE_HOME
```

On a Windows operating system, if your Oracle home directory is `C:\Oracle\Product\ORACLE_HOME`, use a file manager window and navigate to the `C:\Oracle\Product` directory, then right-click on the `ORACLE_HOME` folder and select **Delete**.

Removing the Program Shortcuts on Windows Operating Systems

On Windows operating systems, you must also manually remove the program shortcuts; the deinstaller does not remove them for you.

To remove the program shortcuts on Windows:

1. Go to the `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Oracle\`*`ORACLE_HOME`*`\`*`Product`* directory.
2. If you only have one product installed in your Oracle home, delete the *`ORACLE_HOME`* directory. If you have multiple products installed in your Oracle home, delete all products before deleting the *`ORACLE_HOME`* directory.

Oracle® Fusion Middleware Installing and Configuring Oracle Stream Analytics, 12c (12.2.1.2)
E77599-02