

# **OSS Analytics Foundation**

Installation, Configuration and Administration Guide

Version 1.1.4

Edition 1.0 – December 2016



**Hewlett Packard**  
Enterprise

# Notices

---

## Legal notice

© Copyright 2016, Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

## Trademarks

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

Java™ is a trademark of Oracle and/or its affiliates.

Microsoft®, Internet Explorer, Windows®, Windows Server®, and Windows NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

HPE Vertica™, the HPE Vertica Analytics Platform™ are trademarks of Hewlett-Packard

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

UNIX® is a registered trademark of The Open Group.

Red Hat® is a registered trademark of the Red Hat Company.

Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

JBoss®, Wildfly are registered trademarks of RedHat Inc.

# Contents

<b>Notices</b> .....	<b>1</b>
<b>Preface</b> .....	<b>6</b>
About this guide.....	6
Audience.....	6
Software versions.....	6
Typographical Conventions.....	6
Associated Documents.....	7
Support.....	7
<b>Chapter 1 Product overview</b> .....	<b>8</b>
1.1 OSSA Server introduction.....	8
1.1.1 Data Mart REST API.....	8
1.1.2 Batch system.....	8
1.2 OSSA Server architecture.....	8
1.2.1 Data Mart REST API architecture.....	9
1.2.2 Batch system architecture.....	10
<b>Chapter 2 Installation prerequisites</b> .....	<b>12</b>
2.1 Hardware and operating systems prerequisites.....	12
2.2 Software prerequisites.....	12
2.2.1 Java.....	12
2.2.2 VERTICA Database Client.....	12
2.2.3 Timezone settings.....	13
2.3 Installation package.....	13
2.4 Code signing.....	13
2.4.1 Signature verification.....	13
2.4.1.1 Import HPE public key.....	13
2.4.1.2 OSSA kit signature verification.....	14
<b>Chapter 3 Product installation</b> .....	<b>15</b>
3.1 Create the Linux User for OSSA Server.....	15
3.2 Install the RPM kit.....	15
3.3 Create the OSSA database, user and schema.....	15
3.4 Configure Data Mart REST API Vertica resource pool.....	16
3.5 Configure the installation.....	16
3.6 Finalize the installation.....	18
3.7 Validate the installation.....	19
<b>Chapter 4 Product configuration</b> .....	<b>20</b>
4.1 Configure OSSA Server as a Red Hat Linux service.....	20
4.2 Security.....	20
4.2.1 Secure the server.....	20
Here is an example of OSS solution deployment:.....	20
4.2.2 Activate SSL.....	21
4.2.3 Cryptographic algorithms.....	22

4.2.4 Enable auditing.....22

4.2.5 Authentication / Single Sign-On.....22

4.2.6 Role based access control.....22

4.2.7 Encryption.....22

4.2.8 DEBUG traces logging.....23

**Chapter 5 Product Administration .....24**

5.1 Start OSSA Server.....24

5.2 Stop OSSA Server .....24

5.3 Check server process.....24

5.4 Load metadata.....24

5.5 Configure ossa database user for your own solution .....25

5.6 Automatic cleanup of OSSA internal tables related to batch jobs execution.....25

5.7 Load a batch job.....27

5.8 Load batch configurations contained in a directory.....27

5.9 Run a batch job.....28

5.10 OSSA Server Admin console .....28

    5.10.1 Batch monitoring.....28

    5.10.2 Batch scheduling administration.....29

**Chapter 6 Product Troubleshooting .....31**

6.1 Logging.....31

6.2 Common errors.....32

    6.2.1 Deployment error about connections.....32

**Chapter 7 Uninstall the OSSA Server.....33**

**Chapter 8 UOC OSSA Add-on .....34**

8.1 Installation.....34

8.2 UOC OSSA Add-on kit signature verification.....34

8.3 UOC OSSA Add-on configuration.....34

# List of tables

Table 1: Software versions.....	6
Table 2: Minimum hardware requirements for OSSA Server.....	12
Table 3: <i>ossa.cfg</i> parameters description.....	18
Table 4: OSSAF SSL configuration parameters description.....	21
Table 5: OSSA Internal Tables Cleanup Batch job properties description.....	26

# List of figures

Figure 1: OSSA Server architecture.....	9
Figure 2: Data Mart REST API architecture.....	10
Figure 3: Batch system architecture.....	11
Figure 4: OSSA Server Admin console – Batch job monitoring.....	29
Figure 5: OSSA Server Admin console – Batch Administration.....	29
Figure 6: OSSA Server Admin console – Batch scheduling.....	30

# Preface

---

## About this guide

---

This guide describes how to install, configure, administrate and troubleshoot the HPE OSS Analytics Foundation software component.

Software component name: HPE OSS Analytics Foundation

Software component version: 1.1.4

Kit name: `ossa-server-1.1.4-MP.noarch.rpm`

## Audience

---

This installation and configuration guide is for anyone who is responsible for installing, uninstalling, configuring, administrating, or troubleshooting the HPE OSS Analytics Foundation.

## Software versions

---

**The terms Unix and Linux are used as a generic reference to the operating system, unless otherwise specified. The software versions referred to in this document are as follows:**

**Table 1: Software versions**

Product version	Supported operating systems
HPE OSS Analytics Foundation version 1.1.4	Red Hat Enterprise Linux Server release 6.8
HPE Vertica version 7.2.3	Red Hat Enterprise Linux Server release 6.8

## Typographical Conventions

---

*Courier Font:*

- Source code and examples of file contents.
- Commands that you enter on the screen.
- Pathnames
- Keyboard key names

*Italic Text:*

- Filenames, programs and parameters.
- The names of other documents referenced in this manual.

**Bold Text:**

- To introduce new terms and to emphasize important words.

## Associated Documents

---

The following documents contain useful reference information:

*HPE OSS Analytics Foundation Release Notes*

*HPE OSS Analytics Foundation Integration Guide*

## Support

---

Please visit our HPE Software Support Online Web site at <https://softwaresupport.hpe.com> for contact information, and details about HPE Software products, services, and support.

The Software support area of the web site includes the following:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information



# Chapter 1 Product overview

---

The HPE OSS Analytics Foundation is part of the HPE OSS Analytics program. It provides a unified solution for external applications to access data and to perform analysis on data stored in data warehouses.

The HPE OSS Analytics Foundation is composed of:

- OSSA Server: a JEE application server (JBoss/Wildfly)
- A JEE application that can be deployed in the application server and provides the following functions:
  - Data Mart REST API
  - Batch system

## 1.1 OSSA Server introduction

---

### 1.1.1 Data Mart REST API

The Data Mart REST API provides generic access to data warehouses.

The genericity of the Data Mart REST API comes from the usage of “metadata”.

Metadata allows one to describe the star schema for any set of database tables that implement a data mart. Concretely; the set of **Facts** and referenced **Dimensions** are described. Metadata can also specify the data aggregations available.

The possible aggregation types are:

- Pre-calculated
- On-the -fly

Once the metadata file is loaded into the OSSA Server, the server is able to reply to Dimensions-and-Facts client requests in order to perform multidimensional analysis.



**IMPORTANT:** HPE OSS Analytics Foundation does not provide any metadata instantiation. Please refer to your domain specific OSS Analytics solution documentation for domain specific metadata.

---

### 1.1.2 Batch system

The Batch system aims at defining, scheduling, monitoring batch jobs.

What is a “batch job” in the context of OSS Analytics?

This is a process that generally handle and transform large amount of data, and for which you want an automatic and periodical execution.

The Batch system allows you to define your batch jobs based on some generic pre-defined but customizable tasks. Those tasks allow you to perform: loading of data to Vertica, SQL processing, summarization on data, or report generation and mailing.

For details about those tasks please refer to the *HPE OSS Analytics Foundation Integration guide*.

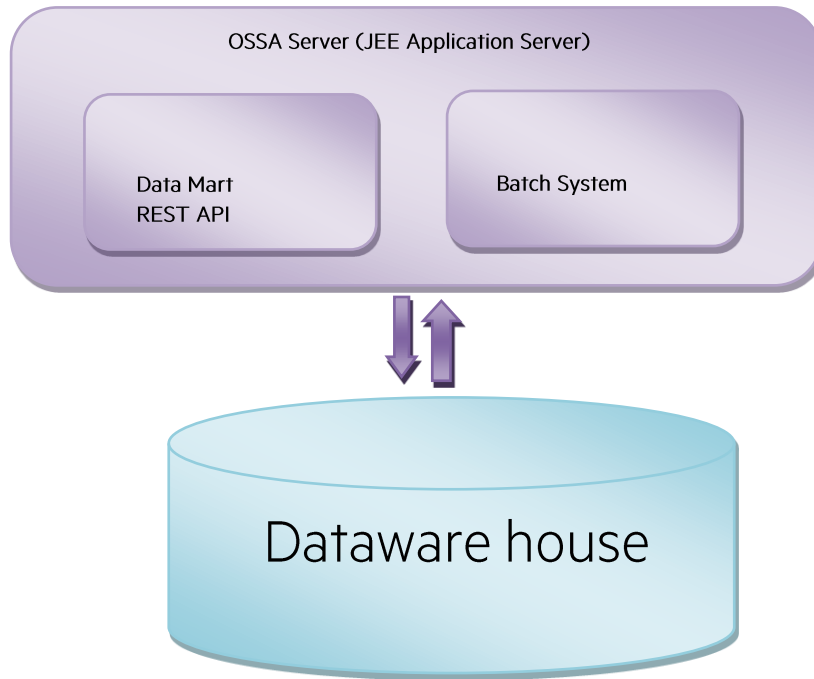
Then, the Batch system allows you to deploy, schedule and monitor your batch jobs.

## 1.2 OSSA Server architecture

---

In OSSA Server a JEE application providing the following functions can be deployed:

- Data Mart REST API
- Batch system



**Figure 1: OSSA Server architecture**

## 1.2.1 Data Mart REST API architecture

The OSSA Server provides a HTTP/HTTPS REST API for external applications to access and compute the required information stored in a data warehouse.

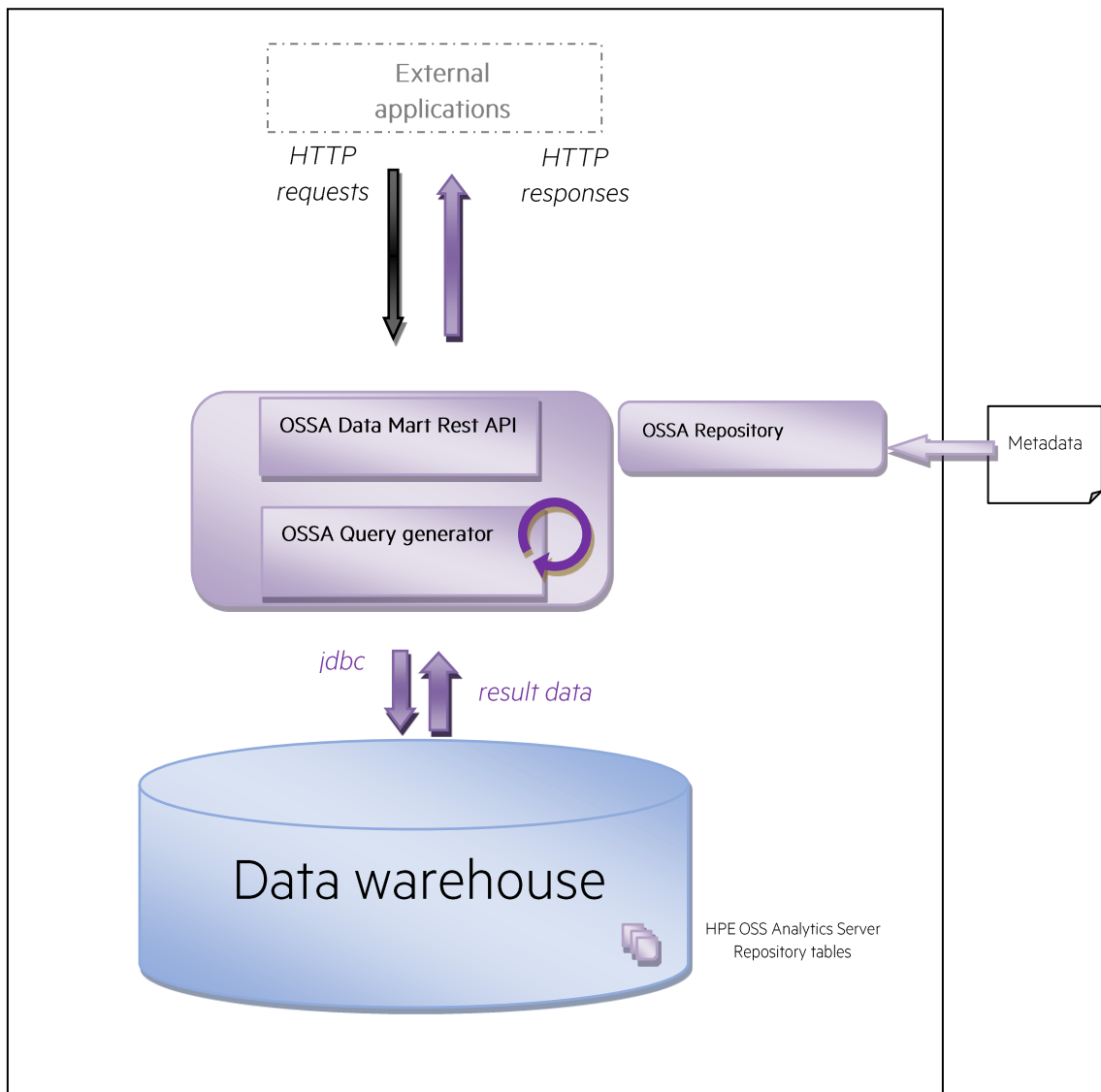
The main operations done by the Data Mart REST API are:

- Analyze the HTTP request
- Generate the corresponding SQL query, by using the domain specific metadata
- Execute the SQL query on the data warehouse
- Retrieve the set of data requested
- Send the HTTP response

The Data Mart REST API consists of:

- REST API and query generator, deployed in a JEE Container (JBoss/Wildfly)  
(Note that JBoss/Wildfly is embedded within the software kit)
- Repository loader tool

The Repository loader tool allows to load metadata, which is then stored in the database, in a dedicated internal ossa schema.



**Figure 2: Data Mart REST API architecture**

## 1.2.2 Batch system architecture

The Batch system relies on the standard JEE Batch API (Batch Applications for the Java Platform, JSR-352).

The main operations offered by the Batch system are:

- Schedule batch jobs whose schedule descriptions are loaded into the repository
- Execute the batch jobs whose processing descriptions are also loaded into the repository

The definitions and executions of the batch jobs rely on OSSA *batchlets* that provide several kind of data processing capabilities: Load data to Vertica, SQL execution, summarization of data, report generation, mailing...

For details about OSSA *batchlets* please refer to the *HPE OSS Analytics Foundation Integration guide*.

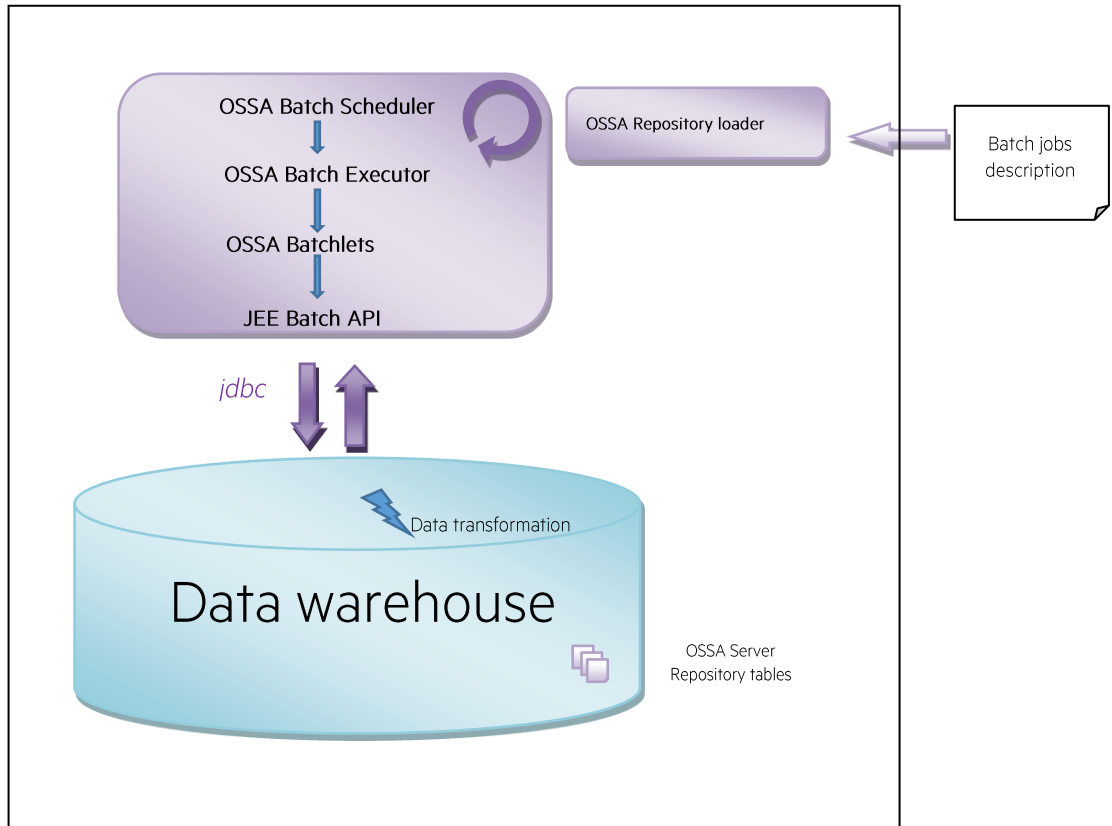


Figure 3: Batch system architecture

## Chapter 2 Installation prerequisites

### 2.1 Hardware and operating systems prerequisites

Before installing OSSA Server, verify that your system meets the following minimum requirements.

**Table 2: Minimum hardware requirements for OSSA Server**

Hardware	Minimum requirement
CPU	1 CPU 2.5 GHz 4 cores
RAM	8 GB
Hard disk size	50 GB
Network	100 MB Ethernet



**NOTE:** The listed minimum requirements are for OSSA Server only (Data Mart REST API and Batch system). This does not include resources that would be required to run any other application of the HP OSS Analytics solution.

Contact the product manager for further details on the sizing requirements.

### 2.2 Software prerequisites

#### 2.2.1 Java

OSSA Server requires *java1.8.0-openjdk*.

Generally, Red Hat Enterprise Linux Server comes with OpenJDK Java VM, but please check if it is installed.

As root user, run:

```
# rpm -qa | grep 1.8.0-openjdk
```

If *java1.8.0-openjdk* is not installed, install it.

As root user, run:

```
# yum update java-1.8.0-openjdk
```

For trouble shooting it is also highly recommended to also install *java1.8.0-openjdk-devel*.

#### 2.2.2 VERTICA Database Client

OSSA Server can be used to retrieve domain-specific data from a Vertica data warehouse. In this context, OSSA Server requires the HPE Vertica client to be installed in order to access the data.

OSSA Repository information is also stored as VERTICA tables.

VERTICA version required is Vertica 7.2.3.

Check if Vertica client is already installed. As root user, run:

```
# rpm -qa | grep vertica-client
```

If not installed, download Vertica client 7.2.3 from the HP Vertica website and install it.

If you need to install VERTICA server to host your data warehouse, see the HPE Vertica Installation Guide (<http://www.vertica.com/documentation/hp-vertica-7-2-x-documentation>) for information on how to install your Vertica database on a given cluster of servers.



**NOTE:** The number of concurrent users that can be served in a suitable response time by OSSA Server depends on the ability of the Vertica cluster to serve the simultaneous requests. It is recommended to build a Vertica platform aligned with the Vertica hardware guide and the system settings described in the Vertica installation and administration guides.



**IMPORTANT:** When installing Vertica, please consider the security guidance, in “*Security and Authentication*” chapter of Vertica documentation:

[https://my.vertica.com/docs/7.2.x/HTML/index.htm#Authoring/Security/ImplementingSecurity.htm%3FTocPath%3DSecurity%2520and%2520Authentication%7C\\_\\_0](https://my.vertica.com/docs/7.2.x/HTML/index.htm#Authoring/Security/ImplementingSecurity.htm%3FTocPath%3DSecurity%2520and%2520Authentication%7C__0)

## 2.2.3 Timezone settings



**CAUTION:** While configuring your platform please ensure that the **local timezone** is set for the operating system and for the java virtual machine and for the Vertica database.

## 2.3 Installation package

OSSA Server software component is delivered as a Red Hat Enterprise Linux RPM package:

```
ossa-server-1.1.4-MP.noarch.rpm
```

## 2.4 Code signing

The code signing procedure allows you to assess the integrity of the delivered product before installing it, by verifying the signature of the software packages.

HPE recommends using signature verification on its products.

### 2.4.1 Signature verification

#### 2.4.1.1 Import HPE public key

Perform the following steps to import the public key that is needed for verifying the delivered products. These steps requires to be logged as root.

1. Create a temporary directory where the public keys will be stored:

```
# mkdir -p signcheck
```

2. Download the public keys:

```
# wget -P signcheck/ https://ftp.hp.com/pub/keys/HPE-GPG-Public-Keys.tar.gz
```

3. Unzip and untar the downloaded file *HPE-GPG-Public-Keys.tar.gz*

4. Import the public key for rpm

```
# rpm --import signcheck/2BAF2262.pub
```

## 2.4.1.2 OSSA kit signature verification

To verify the integrity of the delivered product, perform the following steps:

```
$ rpm -Kv ossa-server-1.1.4-MP.noarch.rpm
```

Check the command output. If signature verification completed successfully, the command output will be:

```
ossa-server-1.1.4-MP.noarch.rpm:  
Header V3 RSA/SHA256 Signature, key ID 9bf2b0ca: NOKEY  
Header SHA1 digest: OK (9be2ae4006b824401f680f18fd38babdbb74b21b)  
V3 RSA/SHA256 Signature, key ID 9bf2b0ca: NOKEY  
MD5 digest: OK (63c9b8f037e0d1dc7342208bdbe5773e)
```



**NOTE:** For more information about signature verification procedure, please visit:

<https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning2>

---

## Chapter 3 Product installation

### 3.1 Create the Linux User for OSSA Server

You need to have root credentials for installing the package; but, the installed files are owned by the standard user, and no processes must run under the root account.

For security reasons, no Linux user is created automatically during the installation process. Therefore, the required user must be created manually prior to the installation.

1. To create the 'ossa' OSS Analytics Foundation Linux user, run the following commands as root user:

```
# useradd -g users -b /home -m -s /bin/bash -c 'HPE OSS Analytics user' ossa
```

2. Modify the password of the ossa user:

```
# passwd ossa
```

### 3.2 Install the RPM kit

You need to have root credentials for installing the package; but, the installed files are owned by the standard user, and no ossa processes must run under the root account.

Install the OSSA Server package on your Linux system.

As *root* user, enter:

```
# rpm -ivh ossa-server-1.1.4-MP.noarch.rpm
```



**NOTE:** The default installation location of the RPM package is `/opt/ossa/`. Use the standard `rpm --prefix` options for installing OSSA in another directory of your choice.

### 3.3 Create the OSSA database, user and schema

The OSSA repository can be used to store in the database configuration parameters for applications running inside OSSA Server and metadata to drive the Data Mart REST API. A dedicated ossa database user and a dedicated ossa schema are used to access and contain all the associated database tables.

The OSSA repository is in the same database as the data warehouse for application domain data.



**IMPORTANT:** In case of upgrade of OSS Analytics Foundation, you do not need to re-create the OSSA schema.

1. If your data warehouse is not yet created, create the database following the Vertica documentation:

*Create a Database Using Administration Tools*

<http://my.vertica.com/docs/7.2.x/HTML/index.htm#Authoring/AdministratorsGuide/ConfiguringTheDB/CreateADatabaseUsingAdministrationTools.htm>



**NOTICE:** The default database name used in configuration file and scripts is: "OSSA".





**NOTICE:** In the following, Vertica `vsq` command examples use the `-w` command line option which displays the database password in plain text on the screen. Use it with care, particularly if you are connecting as the database administrator.

2. Ensure that the Vertica `vsq` client software is in the `ossa` Linux user `PATH` environment variable.
3. Create the OSSA database user within the database created previously.  
Note that if your database is not named 'OSSA', then in `/opt/ossa/dcl/create_user_ossa.sql` replace  

```
DATABASE OSSA with DATABASE <yourDatabaseName>
```

For creating the `ossa` DB user, enter as `ossa` Linux user:

```
$ cd /opt/ossa/dcl
$ vsq -d <DBname> -h <DBIPaddress> -U <DBAdministratorUserName> -w <DBAdministratorPassword> -v
  ossa_user=ossa -f create_user_ossa.sql
```

Note that, with this command, a default Vertica dedicated resource pool `pool_ossa` is created (please refer to next section for configuring it)

4. Create the `ossa` schema for this `ossa` database user.  
Enter as `ossa` Linux user:

```
$ vsq -d <DBname> -h <DBIPaddress> -U ossa -w ossapwd -v ossa_user=ossa -f
  create_schema_ossa.sql
```

**NOTICE:** `ossapwd` is an example of a password. Obviously, you can consider choosing another much stronger one.

5. Verify that the schema was created successfully by running the following command as `ossa` Linux user.  
You should get 0 rows with empty table.

```
$ vsq -d <DBname> -h <DBIPaddress> -U ossa -w ossapwd -c 'select id from package'
```

## 3.4 Configure Data Mart REST API Vertica resource pool

The Data Mart REST API is usually not the only application accessing the Vertica database. In order to separate the different access loads to the database and to configure them differently, a dedicated **Vertica resource pool** for Data Mart REST API is available (`pool_ossa` has been created with the previous steps)

This resource pool can be configured according to VERTICA resources available and number of users that are expected to use the Data Mart REST API.

**Example:**

```
$ vsq -d <DBname> -h <DBIPaddress> -U <DBAdministratorUserName> -w <DBAdministratorPassword>
OSSA=> ALTER RESOURCE POOL POOL_OSSA MEMORYSIZE '10G' MAXMEMORYSIZE '10G' EXECUTIONPARALLELISM 1
      MAXCONCURRENCY 32 PLANNEDCONCURRENCY 32 RUNTIMEPRIORITY LOW RUNTIMEPRIORITYTHRESHOLD 0 PRIORITY
      0;
```

For more information on resource pool settings please refer to Vertica documentation:

[https://my.vertica.com/docs/7.2.x/HTML/index.htm#Authoring/SQLReferenceManual/SystemTables/CATALOG/RESOURCE\\_POOLS.htm](https://my.vertica.com/docs/7.2.x/HTML/index.htm#Authoring/SQLReferenceManual/SystemTables/CATALOG/RESOURCE_POOLS.htm)

## 3.5 Configure the installation

All remaining steps must be run with the 'ossa' unix user.

As *ossa* user, edit the */opt/ossa/ossa.cfg* installation configuration file: modify the parameters according to your environment.



---

**NOTE:** In case you are installing OSSAF for the first time, you can create this *ossa.cfg* configuration file by taking a default example delivered. You can:

```
cp /opt/ossa/ossa.cfg.dist /opt/ossa/ossa.cfg
```

and then modify the *ossa.cfg*

---

The expected values for the parameters are the following:

**Table 3: ossa.cfg parameters description**

Parameter	Expected value
<i>JBOSS_HOST</i>	The address to which JBoss binds. The default value is <i>localhost</i> . If you want to bind JBoss to the IP address of the host, you must modify the value to <code>`hostname -i`</code>
<i>JBOSS_PORT_OFFSET</i>	The value to add to the default http port number (8080) for defining a different port on which JBoss runs. Ex: value 100 makes JBoss web server running on port 8180. The default value is 0, meaning that JBoss runs on default port 8080.
<i>JBOSS_HTTP_PORT</i>	Do not modify this parameter. It calculates automatically the requested JBoss HTTP port based on the <i>JBOSS_PORT_OFFSET</i> you have set.
<i>JBOSS_JAVA_OPTS</i>	The Java options to pass to the JBoss server. -Xmx must be set according to the memory requirement for all the applications deployed in the OSSA Server. Keep the default values for standard usage.
<i>JBOSS_SERVER_OPTS</i>	Do not modify this parameter. It specifies OSSA Server 'node' name that is used to identify different OSSA Server for high availability purpose.
<i>OSSA_DB_TYPE_01</i>	The data warehouse database type. In this version of OSSA Server only "vertica" is supported.
<i>OSSA_DB_HOST_01</i>	The hostname or IP address of the database server hosting the data warehouse.
<i>OSSA_DB_NAME_01</i>	The name of the database hosting the data warehouse.
<i>OSSA_DB_PORT_01</i>	The port of the database.
<i>OSSA_DB_USER_01</i>	The name of the database user to access the data warehouse.
<i>OSSA_DB_PASSWORD_01</i>	The password of the database user to access the data warehouse.
<i>OSSA_DB_URL_01</i>	In case database connection needs some extra parameters, you can fully specify it with this parameter. Example, for setting load balancing, you can specify: "jdbc:\${OSSA_DB_TYPE_01}://\${OSSA_DB_HOST_01}:\${OSSA_DB_PORT_01}/\${OSSA_DB_NAME_01}?BackupServerNode=mynode1.mydomain.com,mynode2.mydomain.com&ConnectionLoadBalance=1"
<i>OSSA_DS_NAME_01</i>	The name of the OSSA Server JEE data source to use in the JEE container. Keep the default values for standard usage.
<i>OSSA_DS_MAX_POOL_SIZE_01</i>	The maximum number of connections created in the OSSA Server JEE connection pool for accessing the data warehouse (for Data Mart REST API purpose).
<i>OSSA_SSL_ENABLED</i>	Y or N. The default value is N, meaning that SSL (TLS) is not enabled. You do not have to modify the other <i>OSSA_SSL_xxx</i> parameters if you do not enable SSL (TLS). For enabling and configuring SSL correctly, see section <a href="#">4.2.1</a>
<i>OSSA_MAIL_SERVER</i>	(optional) The address of the mail server you want to declare in JBoss configuration.
<i>OSSA_MAIL_PORT</i>	(optional) The port of the mail server you want to declare in JBoss configuration.

Once you have checked that all the listed parameters have correct values, use the defined work environment by sourcing the *ossa\_env.sh* file:

```
$ source /opt/ossa/bin/ossa_env.sh
```

## 3.6 Finalize the installation

Run the *ossa-config.sh* tool in order to automate the configuration of the server based on the parameters you have previously defined:

```
$ ${OSSA_HOME}/bin/ossa_config.sh
```

OSSA server is now configured. You can then start the server:

```
$ jbossstart
```



**CAUTION:** In case you modify *ossa.cfg* the OSSAF server must be re-configured, so please, stop OSSAF server, and execute: `${OSSA_HOME}/bin/ossa_config.sh`

## 3.7 Validate the installation

1. Open an internet browser and check the following URL:

```
http://${JBOSS_HOST}:8080/
```



**NOTE:** `${JBOSS_HOST}` is the hostname or IP address of the host on which you installed OSSA Server. 8080 is the default port used for the server. You may have modified this port by setting an offset at configuration time (section 3.4, parameter `JBOSS_PORT_OFFSET`).

This must display the OSS Analytics Foundation Administration Console and this confirms that OSSA Server is running.

2. Then, type this other URL:

```
http://${JBOSS_HOST}:8080/ossa/packages/
```

This checks that Data Mart Rest API is running. The output gives you the list of metadata packages loaded into the server.

If no metadata packages are loaded, the result will simply be: [ ]

If those checks give correct results, you are now ready to load your domain-specific metadata, domain-specific batch jobs.

## Chapter 4 Product configuration

### 4.1 Configure OSSA Server as a Red Hat Linux service

To configure OSS Analytics Foundation as a RedHat Linux service in order to have it automatically started at boot time, and stopped at halt time, as root user, execute:

```
chkconfig --add ossa-server
```

## 4.2 Security

### 4.2.1 Secure the server

By default, secured communication across the components of the server is not enabled, as it requires first the administrator to install the certificate of the platform for the SSL authentication. You can activate SSL as described in next section, to protect private data that might be published by the deployed specific domain metadata.



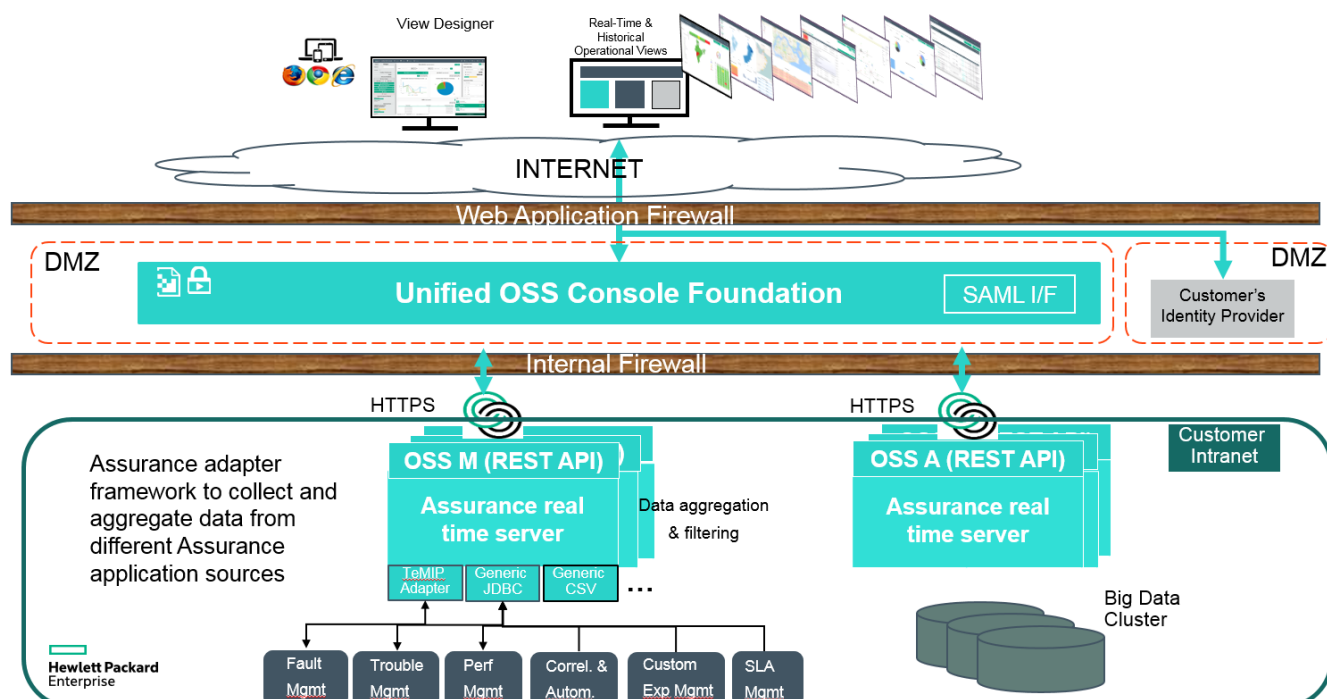
**CAUTION:** All data sent to the OSSA REST APIs is assumed to be trusted data, the data is not checked for malicious content and that it is assumed that the OSS presentation layer validates all untrusted data before using the REST APIs.



**NOTICE:** Recommendations concerning security aspects of a full OSS solution deployment can be found in the OSS hardening guide. Please ask OSS solution manager if needed.

Here is an example of OSS solution deployment:

### HPE OSS Analytics deployment in a protected environment



## 4.2.2 Activate SSL

This process is optional but the OSSA Server can be configured with server authentication and secured communication, with the use of HTTPS.



**NOTICE:** This feature deals with the authentication of the OSSA Server for the web clients. It does not aim at authenticating web clients. For authenticating web clients, your user interface application has to rely on an independent identity provider, specific to your company that manages the users.

1. As ossa user, edit the `/opt/ossa/ossa.cfg` installation configuration file.
2. Modify the parameters according to your environment.

For test or demonstration purposes, you can configure the OSSA Server with an example of SSL certificate, which is automatically generated during configuration.

- a. During the configuration, modify the `OSSA_SSL_ENABLED` parameter value, in `ossa.cfg`, to Y.
- b. Keep the default values for the other `OSSA_SSL_xxx` parameters.  
In the application server, a self-signed certificate is automatically generated with a fake organizational identity and location. The generated certificate is an example that can be used for demonstration purposes only.

On production systems, you must include your company SSL certificate (Java keystore file) and define accordingly the `OSSA_SSL_xxx` parameters.

Check the expected values carefully for the parameters:

**Table 4: OSSAF SSL configuration parameters description**

Parameter	Expected value
<code>OSSA_SSL_ENABLED</code>	Y or N. Set it to Y for enabling SSL.
<code>OSSA_SSL_KEYTOOL_PATH</code>	The path to the <i>keytool</i> utility. <i>keytool</i> is the cryptographic key and certificate management utility which is installed with <i>java-1.8.0-openjdk</i> .
<code>OSSA_SSL_JAVA_KEYSTORE_FILE</code>	The location of your company SSL certificate; a Java <i>keystore</i> file containing keys and the certificate. (your 'ossa' linux user must have read access on this file)
<code>OSSA_SSL_KEYSTORE_ENTRY_ALIAS</code>	The logical name of the entry within the <i>keystore</i> file, where the certificate and the private key are stored.
<code>OSSA_SSL_KEYSTORE_PASSWD</code>	The password of the Java <i>keystore</i> file.

3. Use the work environment with the configured parameters, by sourcing the `ossa_env.sh` file:

```
$ source /opt/ossa/bin/ossa_env.sh
```

4. Finalize the configuration of the OSSA Server (see section [3.5](#))

With those settings, for accessing the OSSA Server, you must use the HTTPS protocol, and use the corresponding port number. The default https port number is 8443 (if you did not modify the value of the `JBOSS_PORT_OFFSET` parameter).

## 4.2.3 Cryptographic algorithms

Using SSL, the certificates are managed through the regular configuration of Java SE 8 JCE keystores. If you use the automatically generated, self-signed example certificate, the encryption is RSA, and the key length is 2048. By using your company SSL certificate, the encryption algorithm and key length can differ.

## 4.2.4 Enable auditing

Security event audit logging is disabled by default, because it may introduce a performance impact on servers. This feature has to be configured to be available on-demand only.

1. To enable it, in the `_${JBOSS_HOME}/standalone/configuration/standalone.xml` file, set the following values:

```
<audit-log>
...
<logger enabled="true" log-boot="true" log-read-only="false">
...
</audit-log>
```

2. Restart the server. The audit log is then available at:
   
`_${JBOSS_HOME}/standalone/data/audit-log.log`



**CAUTION:** These logs must be accessible only to platform administrator, because they may contain sensitive privacy information, for example the IP address.

## 4.2.5 Authentication / Single Sign-On

The OSSA Server does not provide a client authentication mechanism.

If your external user interface application enables integration with identity providers through the SAML protocol, and you want the application server from OSSA Server to host an identity provider, please follow the corresponding documentation from JBoss Wildfly 10.

## 4.2.6 Role based access control

The OSSA Server Data Mart REST API provides role based access control (RBAC).

The definitions of the different roles are defined within each specific domain metadata. The definitions determine which roles can access which data.

The definition and use of roles are optional.

The Data Mart REST API analyzes the user roles sent in the request URLs and compares them to the ones specified within the metadata for the requested data; and then determines whether the data can be accessed or not.

## 4.2.7 Encryption

The Vertica JDC driver does not encrypt the database connection password. If the platform deployment exposes the dedicated private network, the IP connection between the OSSA server and the underlying Vertica database, it is strongly recommended to configure IPsec for that transport, to secure the connection.

## 4.2.8 DEBUG traces logging

If troubleshooting requires to enable the DEBUG traces logging, some private, and confidential data recorded in the database can appear in the log files. Access to these files must be limited to the authorized users. Debug trace logging is to be managed with a system login account.



## Chapter 5 Product Administration

---

### 5.1 Start OSSA Server

---

As ossa user, run:

```
$ source /opt/ossa/bin/ossa_env.sh
$ jbossstart
```

### 5.2 Stop OSSA Server

---

As ossa user, run:

```
$ source /opt/ossa/bin/ossa_env.sh
$ jbossstop
```

### 5.3 Check server process

---

As ossa user, run:

```
$ source /opt/ossa/bin/ossa_env.sh
$ jbossshow
```

If OSSA server is running, the output will display process information: pid, command.

If OSSA server is not running, the output will be empty.

### 5.4 Load metadata

---

1. Check the OSS Analytics solution you have chosen in order to retrieve your domain specific metadata. This metadata contains information about:
  - Description of the data to analyze
  - Database location and tables and columns that contains the data
  - Type of aggregations that can be performed on the data

This metadata file must be loaded into OSSA Server with the use of the repository loader.

2. Make sure that:
  - OSSA Server is running
  - Metadata specific database schema is loaded
3. In the terminal where you sourced *ossa\_env.sh*, as ossa user, load your metadata:

```
$ ${OSSA_HOME}/bin/ossa-repo.sh loadMetadata <full_path_to_metadata_file>
$ ${OSSA_HOME}/bin/ossa-repo.sh reload
```

Output:

```
loadMetadata <metadata file>
OK
```

- Then, you can start using the Data Mart REST API for your specific domain, through your user interface application.

Depending on the OSS Analytics solution you are deploying you could use the HPE Unified OSS Console v2.x with its HPE OSS Analytics add-on.

To load the metadata and the associated HPE OSS Console v2 views and workspaces JSON definition files, in the terminal where you sourced `ossa_env.sh`, as `ossa` user, execute:

```
$ ${OSSA_HOME}/bin/ossa-repo.sh loadMetadataViewsWks <full_path_of_metadataFile.xml>
  <full_path_of_UOC_ViewsFile.json> <full_path_of_UOC_WorkspacesFile.json>
$ ${OSSA_HOME}/bin/ossa-repo.sh reload
```

## 5.5 Configure ossa database user for your own solution

In case you integrate your own Analytics solution, with your own metadata, with your own database schema/tables, you must configure the ossa database user accordingly.

Please perform the following commands in order to grant ossa database user access to your data.

Connect to the database with your own database user and perform:

```
GRANT USAGE ON SCHEMA <your_own_schema> TO <ossaDBUser>;
GRANT SELECT ON ALL TABLES IN SCHEMA <your_own_schema> TO <ossaDBUser>;
```

in case you have defined some Vertica views which are referenced in your metadata:

```
GRANT SELECT ON <your_own_vertica_view> to <ossaDBUser>;
```

in case you have defined specific functions which are referenced in your metadata:

```
GRANT EXECUTE ON FUNCTION <your_own_db_user>.<your_own_function>(,..,..,..) TO
  <ossaDBUser>;
```

Finally, connect to the database as dbadmin user and perform:

```
alter user <ossaDBUser> search_path <ossaSchema>, <your_own_schema>, PUBLIC,
  v_catalog, v_monitor, v_internal;
exit;
```

## 5.6 Automatic cleanup of OSSA internal tables related to batch jobs execution

Batch jobs execution related history and information are recorded in OSSA internal tables (`JOB_INSTANCE`, `JOB_EXECUTION`, `STEP_EXECUTION`, `PARTITION_EXECUTION`, and `BATCH_CDC_CURSOR`). These internal tables will get larger as all batch jobs execute continuously and periodically.

Two batch jobs are delivered to clean up these internal tables; by deleting job execution histories which exceed the retention policy, to keep their sizes balanced.

- `OSSAFcleanupJBeretTables`

`${OSSA_HOME}/repo-ossa/cleanup/OSSAFcleanupJBeretTables.Job.xml`  
`${OSSA_HOME}/repo-ossa/cleanup/BATCH_OSSAFcleanupJBeretTables.json`

- `OSSAFcleanupBatchCdcCursor`

`${OSSA_HOME}/repo-ossa/cleanup/OSSAFcleanupBatchCdcCursor.Job.xml`  
`${OSSA_HOME}/repo-ossa/cleanup/BATCH_OSSAFcleanupBatchCdcCursor.json`

The .xml files contain the applicative processing of the batch job and retention policy configuration.

The .json files contain the execution description of the batch job: job parameters, administrative state (lock or unlocked) and batch schedule.

Here are the configuration parameters of those batch jobs, in case you want to change them.  
 For a “standard” usage of OSSA, you can keep default values.

1. Configure the retention policy and schedule job execution:

**Table 5: OSSA Internal Tables Cleanup Batch job properties description**

Property	Job File	Description
jobExecutionRetentionDay	<code>\${OSSA_HOME}/repo-ossa/cleanup/OSSAFcleanupJBeretTables.Job.xml</code>	Batch jobs executed in recent retention days will be kept and the others will be cleaned up. e.g. <pre>&lt;properties&gt; ... &lt;property name="jobExecutionRetentionDay" value="30"/&gt; ... &lt;/properties&gt;</pre>
batchCdcCursorRetentionDay	<code>\${OSSA_HOME}/repo-ossa/cleanup/OSSAFcleanupBatchCdcCursor.Job.xml</code>	The cdc cursor records updated in recent retention days are kept and those exceeding the retention day will be cleaned up (except if the more recent cdc is in the ‘deletion’ period). e.g. <pre>&lt;properties&gt; ...   &lt;property name="batchCdcCursorRetentionDay" value="7"/&gt; ... &lt;/properties&gt;</pre>
batchSchedule	<code>\${OSSA_HOME}/repo-ossa/cleanup/BATCH_OSSAFcleanupJBeretTables.json</code> , <code>\${OSSA_HOME}/repo-ossa/cleanup/BATCH_OSSAFcleanupBatchCdcCursor.json</code>	“dayOfWeek”: Description: One or more days within a week. Default value: * Allowable values: 0 to 7 (both 0 and 7 refer to Sunday) “hour”: Description: One or more hours within a day. Default value: 0 Allowable values: 0 to 23 “minute”: Description: One or more minutes within an hour. Default value: 0 Allowable values: 0 to 59 “second”: Description: One or more seconds within a minute. Default value: 0 Allowable values: 0 to 59 e.g. <pre>"batchSchedule" : {   "dayOfWeek" : "1",   "hour" : "3",   "minute" : "0",</pre>

		<pre>"second" : "0" }</pre> <p>See <a href="#">J2EE Timer Service documentation</a> for more details.</p>
--	--	---

For scheduling those cleanup batch jobs, please follow those steps:

1. Make sure that OSSA Server is running.
2. In the terminal where you sourced `ossa_env.sh`, as `ossa` user, execute:

```
$ ${OSSA_HOME}/bin/ossa_load_cleanupBatchInternalTables.sh
```

Now, the automatic cleanup of OSSA internal tables is in place.

## 5.7 Load a batch job

Batch job can be configured in the Batch system, with two files.

The `<batchJobName>.xml` file contains the applicative processing of the batch job.

The `<BATCH_batchJobName>.json` file contains the execution description of the batch job: job parameters, administrative state (lock or unlocked) and batch schedule.

1. Make sure that OSSA Server is running
2. In the terminal where you sourced `ossa_env.sh`, as `ossa` user, load the description of the processing of the job.

```
$ ${OSSA_HOME}/bin/ossa-repo.sh loadParam <packageName> <jobProcessingName> <BatchJob.xml>
```

Then, load the execution description of the job.

```
$ ${OSSA_HOME}/bin/ossa-repo.sh loadParam <packageName> BATCH_<jobDescriptionName>
  <BATCH_batchJob.json>
$ ossa-repo.sh reload
$ ossa-batch.sh reload
```



### CAUTION:

1. The JobID defined in the ID job attribute within the batch xml definition file should be the same than the batch.  
Example: if a job is defined in the `MyBatch.xml` file, the `job id` in `MyBatch.xml` must be `MyBatch`
2. For the second command, the `<jobDescriptionName>` must start with 'BATCH\_'
3. Within the `BATCH_batchJob.json` file, the `jobXmlPath` value must be identical to the `<jobProcessingName>` that you have put in the first command.
4. On both commands, `<BatchJob>.xml` and `<BATCH_batchJob>.json` must be full paths to files.

If you have loaded an unlocked batch job, the **batch job is automatically scheduled and will start at next schedule time**.

## 5.8 Load batch configurations contained in a directory

This command allows you to load several batch jobs which applicative processing descriptions and execution descriptions are contained in a directory.

In the terminal where you sourced `ossa_env.sh`, as `ossa` user, execute:

```
$ ${OSSA_HOME}/bin/ossa-repo.sh loadDirectory <packageName> <full_path_of_directory>
$ ossa-repo.sh reload
$ ossa-batch.sh reload
```

This loads into the OSSA repository for the *<package\_name>*, the content of the directory:

- "*param\_name*" populated with the name of each file
- "*param\_value*" populated with the content of each file

In that way, with a one shot command, a set of batch jobs *.json* and *.xml* files contained in a directory can be uploaded to the OSSA repository.



**CAUTION:** The same recommendations, as previous section in term of naming of files and content of files must be followed.

## 5.9 Run a batch job

In case you want to launch a one-time execution of your batch job (for testing purpose for example), you have two choices:

- either you run the job asynchronously, meaning that the control is given back to you directly after you enter the command, without waiting for the completion of the batch job

```
$ ossa-batch.sh run <jobName> [<jobParameters.json>]
```

- or you can run the job synchronously, meaning that the command gives you back the control once the job is completed:

```
$ ossa-batch.sh run-sync <jobName> [<jobParameters.json>] [timeout]
```

**BATCH-NAME** (required) name of the batch to be run  
**JSON\_PARAM** (optional) JSON object representing the batch job parameters  
**TIMEOUT** (optional) Maximum time in milliseconds to wait before returning. It won't stop the job but it will simply give back control to caller process.

## 5.10 OSSA Server Admin console

The OSSA Server provides a web console for administration purpose. The main usage of this console is to monitor the execution of the batch jobs.


This administration console is available at the following URL:


```
http://${JBoss_HOST}:${JBoss_HTTP_PORT}/
```

### 5.10.1 Batch monitoring

When clicking on the **Batch Monitor** tab, the list of batch jobs is displayed.

- The **Xml** column allows you to click on each batch job xml in order to display its processing description

- When clicking on the button  at the end of the line, you have access to the history of the batch jobs executions.

- If you want some more details about a job execution, you can click on the button  at the end of the line: this then lists the steps executions of this batch job.

The screenshot displays the OSSA Server Admin console interface for batch job monitoring. It features a navigation bar at the top with tabs for 'Home', 'Configuration', 'Batch Scheduler', 'Batch Monitor', and 'Metadata RestAPI'. The main content area is divided into three sections:

- OSSA Foundation Batch Engine:** A table listing batch jobs with columns for Package, Batch, and Xml. The jobs listed are:
 

Package	Batch	Xml
com.hp.ossa.test.batchlet	TestCopyToVertica	testcasesSql/TestCopyToVertica.xml
com.hp.ossa.test.batchlet	TestSql-01	testcasesSql/TestSql-01.xml
com.hp.ossa.test.batchlet	TestSql-02	testcasesSql/TestSql-02.xml
- Job TestCopyToVertica history:** A table showing the execution history of the 'TestCopyToVertica' job. The columns are Execution ID, Status, Exit, Date, Create, Start, End, and Touch. All jobs shown are 'COMPLETED' with an 'OK' exit status.
 

Execution ID	Status	Exit	Date	Create	Start	End	Touch
1435152069735	COMPLETED	OK	15/06/24	15:21:09	15:21:09	15:21:10	15:21:10
1435137239772	COMPLETED	OK	15/06/24	11:13:59	11:13:59	11:14:01	11:14:01
1434545108095	COMPLETED	OK	15/06/17	14:45:08	14:45:08	14:45:08	14:45:08
1434367225693	COMPLETED	OK	15/06/15	13:20:25	13:20:25	13:20:26	13:20:26
1434366910979	COMPLETED	OK	15/06/15	13:15:10	13:15:10	13:15:11	13:15:11
- Steps:** A table showing the individual steps of a job. The columns are ID, Step Name, Exit Status, Start Time, End Time, and Duration (ms).
 

ID	Step Name	Exit Status	Start Time	End Time	Duration (ms)
1435152069785	init	INIT-DONE	2015-06-24 15:21:09	2015-06-24 15:21:09	75
1435152069914	createDdl	1	2015-06-24 15:21:09	2015-06-24 15:21:10	185
1435152070175	loadData	COMPLETED	2015-06-24 15:21:10	2015-06-24 15:21:10	124

Figure 4: OSSA Server Admin console – Batch job monitoring

## 5.10.2 Batch scheduling administration

The Batch Scheduler tab allows you to display and administrate batch scheduling.

You can see the administrative state of a batch job: *Locked* or *Unlocked*. You can modify it.

You also have access to the scheduling information by pushing the  button.

The screenshot displays the OSSA Server Admin console interface for batch administration. It features a navigation bar at the top with tabs for 'Home', 'Configuration', 'Batch Scheduler', 'Batch Monitor', and 'Metadata RestAPI'. The main content area is titled 'OSSA Foundation Scheduler' and shows a table of batch jobs with columns for Package, Batch, Status, and Scheduling.

Package	Batch	Status	Scheduling		
				Day Of Week	Day Of Month
com.hp.ossa.test.batchlet	testcasesSql/BATCH_TestcaseTestCopyToVertica.json	Locked	* * * * * /1		
com.hp.ossa.test.batchlet	testcasesSql/BATCH_Testcase_testSql-01.json	Locked	* * * * * /1		
com.hp.ossa.test.batchlet	testcasesSql/BATCH_Testcase_testSql-02.json	Locked	* * * * * /1		

Figure 5: OSSA Server Admin console – Batch Administration

Edit batch scheduling
✕

**Day of week** \*

0 to 7 (both 0 and 7 refer to Sunday). For example: dayOfWeek="3".  
Sun, Mon, Tue, Wed, Thu, Fri, Sat. For example: dayOfWeek="Mon".

**Hour** \*

0 to 23. For example: hour="13".

**Day of Month** \*

1 to 31. For example: dayOfMonth="15".  
-7 to -1 (a negative number means the nth day or days before the end of the month). For example: dayOfMonth="-3".  
Last. For example: dayOfMonth="Last".  
[1st, 2nd, 3rd, 4th, 5th, Last] [Sun, Mon, Tue, Wed, Thu, Fri, Sat]. For example: dayOfMonth="2nd Fri".

**Minute**

\*/1

0 to 59. For example: minute="15".

**Month** \*

1 to 12. For example: month="7".  
Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. For example: month="July".

**Year** \*

A four-digit calendar year. For example: year="2011".

**com.hp.ossa.test.batchlet**  
testcasesSql/BATCH\_TestcaseTestCopyToVertica.json :

```
{
  "jobNamePath": "testcasesSql/TestCopyToVertica.xml",
  "jobParameters": {},
  "adminState": "Locked",
  "batchSchedule": {
    "minute": "*/1",
    "dayOfWeek": "*",
    "dayOfMonth": "*",
    "month": "*",
    "year": "*",
    "hour": "*"
  }
}
```

[See J2EE Timer Service documentation](#) for more details.

Cancel
Save

**Figure 6: OSSA Server Admin console – Batch scheduling**

## Chapter 6 Product Troubleshooting

### 6.1 Logging

The log files related to OSSA Server are stored at the following location:

```
${JBoss_HOME}/standalone/log/
```

- `server.log` for all the JBoss logs
- `ossa_server.log` for the specific OSSA Server log files

The current log level used for each application component can be displayed, as `ossa` user, enter the following command in the terminal where you sourced `ossa_env.sh`:

```
$ ossa_server_get_log
```

Output example:

```
[standalone@xx.xx.xx.xx:9990 /] /subsystem=logging/logger=com.hp.ossa.common:read-attribute(name=level)
{
  "outcome" => "success",
  "result" => "INFO"
}
[standalone@xx.xx.xx.xx:9990 /] /subsystem=logging/logger=com.hp.ossa.repo:read-attribute(name=level)
{
  "outcome" => "success",
  "result" => "INFO"
}
[standalone@xx.xx.xx.xx:9990 /] /subsystem=logging/logger=com.hp.ossa.querygen:read-attribute(name=level)
{
  "outcome" => "success",
  "result" => "INFO"
}
[standalone@xx.xx.xx.xx:9990 /] /subsystem=logging/logger=com.hp.ossa.restapi:read-attribute(name=level)
{
  "outcome" => "success",
  "result" => "INFO"
}
```

The *INFO* level is the default setting.

Note that the *DEBUG* level allows you to see, for example, all the requests sent to the server from the different client IP addresses, and the result of each request.

For troubleshooting purposes, you can decide to set a different log level for all the loggers by executing the command:

```
$ ossa_server_set_log <level>
```

Where `<level>` can take the following values, in descending order of detail:

TRACE, DEBUG, INFO, WARN, ERROR, FATAL.



**CAUTION:** Check the disk space consumption when setting levels below INFO.

For fine grain troubleshooting, you can optionally set a level of log on specific loggers. For this, execute this command:



```
$ jbosssetlog <logger> <level>
```

## 6.2 Common errors

### 6.2.1 Deployment error about connections

When starting OSSA Server, if you encounter the following error in server log files, it means that your OSSA Server data source cannot be reached.

Here are the kind of connection errors you can face:

```
Unable to create connection
or
Caused by: com.vertica.support.exceptions.ErrorException
or
Failed to obtain connection
or
Failed to create JDBC job repository
```

1. In order to troubleshoot the problem, as `ossa` user, source your environment:

```
$ source /opt/ossa/bin/ossa_env.sh
```

2. Check if you can connect to your Vertica database using the Vertica client:

```
$ vsql -d ${OSSA_DB_NAME_01} -h ${OSSA_DB_HOST_01} -p ${OSSA_DB_PORT_01} -U ${OSSA_DB_USER_01} -w
${OSSA_DB_PASSWORD_01}
```

If you can connect to the server, it means that there was an environment problem at the time the OSSA Server was started. Here are the possible root causes:

- Your Vertica database is not started. Contact your Vertica database administrator, and restart the OSSA Server.
- The configuration parameters you have set in the `ossa.cfg` are incorrect. Check the correctness of the `OSSA_DB_xxx` parameters carefully. Once they are corrected, follow those steps:
  - Stop OSSA Server
  - Re-run the configuration as described in section [Configure the installation](#)

## Chapter 7 Uninstall the OSSA Server

---

1. In the terminal where you sourced `ossa_env.sh`, as `ossa` user, stop the OSSA Server:

```
$ jbossstop
```

2. As `root` user, uninstall the OSS Analytics Foundation Linux package:

```
# rpm -ev ossa-server-1.1.4-MP.noarch.rpm
```

3. (Optional) If you have configured OSSA Server as a Red Hat Linux service, as `root` user, execute:

```
# chkconfig --del ossa-server
```



---

**IMPORTANT:** The uninstallation does not remove the `ossa.cfg` file, which contains your OSSA configuration. If you plan to install a new version of OSSAF, **keep this file at its current location**. During the next installation, the file will not be overwritten and will be used as it is now. In that way, the same configuration is used for your next installation.

---

## Chapter 8 UOC OSSA Add-on

---

In case you want to have UOCv2 on top of your OSSA server, the software component **UOC OSSA Add-on** is mandatory. It must be installed on the UOC server side.

### 8.1 Installation

---

UOC OSSA Add-on software component is delivered as a Red Hat Enterprise Linux RPM package:

```
uoc-addon-ossa-1.1.4-MP.x86_64.rpm
```

The installation can be done using the standard rpm command:

As *root* user (on the UOC server side) perform the following command:

```
# rpm -ivh uoc-addon-ossa-1.1.4-MP.x86_64.rpm
```

### 8.2 UOC OSSA Add-on kit signature verification

---

The code signing procedure allows you to assess the integrity of the delivered product before installing it, by verifying the signature of the software packages.

HPE recommends using signature verification on its products.

You should already have imported HPE public key (if you have verified signature of OSSA kit – refer to chapter 2.4).

So, you can verify the signature of the UOC OSSA add-on:

```
$ rpm -Kv uoc-addon-ossa-1.1.4-MP.x86_64.rpm
```

Check the command output. If signature verification completed successfully, the command output will be:

```
uoc-addon-ossa-1.1.4-MP.x86_64.rpm:  
Header SHA1 digest: OK (68447843271541e3fb0a3440810d6e4dc642f0ef)  
MD5 digest: OK (d317f6e56223df48dc066d3ccc0372c9)
```



**NOTE:** For more information about signature verification procedure, please visit:

<https://h20392.www2.hpe.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning2>

### 8.3 UOC OSSA Add-on configuration

---

If you use the UOC OSSA Add-on, you need to provide the host and the port of the OSSA server. For this, please edit the following file (JSON syntax):

```
$ vi /opt/uoc2/server/public/addons/plugins/ossa/config.json
```