

SpagoBI JBoss Installation Manual

Authors Fiscato Luca
 Andrea Zoppello
 Zerbetto Davide

Review Grazia Cazzin

Index

1	VERSION.....	3
2	DOCUMENT GOAL	3
3	REFERENCES	3
4	REQUIREMENTS	3
5	INSTALLING SPAGOBI CORE	3
5.1	INSTALL ADDITIONAL LIBRARIES	3
5.2	INSTALLING THE METADATA DATABASE.....	4
5.2.1	<i>Install the database driver</i>	<i>4</i>
5.2.2	<i>Configuring Datasource</i>	<i>5</i>
5.3	INSTALL SPAGOBI PLATFORM.....	5
5.3.1	<i>Configure Hibernate</i>	<i>6</i>
5.4	CMS CONFIGURATION	6
5.5	ENGINES	7
5.5.1	<i>Install SpagoBIJasperReportEngine.....</i>	<i>7</i>
5.5.2	<i>Install SpagoBIJPivotEngine</i>	<i>7</i>
5.6	INSTALL EXAMPLE PORTAL	8

1 Version

Version/Release n° :	1.2	Data Version/Release :	Feb, 3 rd 2006
Update description:	JBoss Installation steps		
Version/Release n° :	1.3	Data Version/Release :	May, 18 th 2006
Update description:	JBoss 4.0.3SP1 server, with eXo Platform v2 Portal 1.1 Final, installation steps		

2 Document goal

This document provides a step by step description for SpagoBI installation and configuration on a JBoss 4.0.3SP1 server with eXo Platform v2 Portal 1.1 Final installed.

3 References

Some of the concepts of this document refer to the following documentation:

- SpagoBI business intelligence platform framework (available at <http://spagobi.eng.it/>)
- Exo Portal Platform (available at <http://www.exoplatform.com>)
- Spago framework (available at <http://spago.eng.it>)
- JBoss application server (available at <http://www.jboss.org>)

4 Requirements

5 Installing SpagoBI Core

We assume that you have correctly installed eXo Platform v2 Portal 1.1 Final on a JBoss 4.0.3SP1 server; in the following we will refer to the exo-JBoss base directory as **EXO-HOME**.

Connect to the SpagoBI's page on the ObjectWeb community site (<http://forge.objectweb.org/projects/spagobi>), click on the 'File' tab, download the file called 'UtilityFiles-bin-1.8.zip', and then unzip it.

The unzip operation produces a folder 'SpagoBIUtilityFiles' which contains 'jboss-server', 'libraries' and other folders. The folder 'jboss-server' respects the tree-folders structure of a clean eXo-JBoss installation; in the following we will refer to this folder as **EXO-INST-FILES**. The folder 'libraries' contains some necessary libraries; in the following we will refer to this as **LIBRARIES**. There's also another folder 'example-portal' which contains a new portal war file useful to install a SpagoBI test portal (However the installation of this example portal is not mandatory); in the following we will refer to this folder as **EXAMPLE_PORTAL**.

5.1 Install additional libraries

SpagoBI needs some additional libraries to manage metadata, so, copy the following libraries from **LIBRARIES** to **EXO-HOME/server/default/deploy/exoplatform.sar**:

- ehcache-1.1.jar
- concurrent-1.3.2.jar
- jackrabbit-0.16.4.1-dev.jar
- jcr-1.0.jar

5.2 Installing the Metadata Database

SpagoBI metadata are stored in a database (for this release SpagoBI supports PostgreSQL, Oracle, MySQL and HSQLDB).

If you don't have anyone of these database servers supported you need to install one of them. Once you have a functional database server you must create a new database for the metadata (spagobi is the database name suggested).

Once completed the operation above it's possible to proceed with the creation and initial population of the metadata database launching the right script for your database server. For each database server supported you need to download from the SpagoBI Repository a zip archive containing the sql script to create the schema, the comments of the table and finally to populate the schema with initial data. (**These scripts don't contain the examples data as the hsqldb**). So, connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file called <<name of you database>>-dbscript-1.4.zip. Into the zip file there are some sql script files, you must execute the creation and insertion script with a client for your database server. In every archive exists also a drop script but this one is useful only if you need to clean your database deleting all the spagobi metadata tables. *Example: if you have a postgresql database server you need to download the relative archive (postgres-dbscript-1.4.zip) and run in the following order the scripts PG_create.sql and PG_insert.sql*

SpagoBI can use also an HSQLDB database, which is very useful for test purpose but is very weak in a production environment. Anyway, if you want to use HSQLDB, just copy the folder **EXO-INST-FILES/sbidata** into **EXO-HOME** and then exec the command `exo-home/sbidata/database/start.bat` (on a windows platform) or `exo-home/sbidata/database/start.sh` (on a unix platform). The command starts an HSQLDB server (listening on port 9002) with a 'spagobi' database already populated with the necessary data.

Every time you start the exo server to work with SpagoBI the database server must be running.

5.2.1 Install the database driver

Before to proceed with persistence configuration we must install the database drivers packages in the application server. Because SpagoBI can be configured to connect to different database servers, one for the metadata and one for datawarehouse for example, you must obtain the specific drivers for every database server used by SpagoBI. The drivers package can be obtained from database vendors site and for the current SpagoBI release we test the following versions:

- Postgresql : postgresql-8.0-311.jdbc2.jar

- Oracle: ojdbc14.jar
- MySQL: mysql-connector-java-3.1.10-bin.jar
- HSQLDB: hsqldb1_8_0_2.jar (contained into **EXO-INST-FILES**/server/default/lib)

The driver jar of the corresponding database you use have to be put under **EXO-HOME**/server/default/lib.

Pay attention: If you decide to use the SpagoBI hsqldb example database add into **EXO-HOME**/server/default/lib the hsqldb1_8_0_2.jar library and remember to remove the existing hsqldb.jar library in the same folder in order to avoid conflicts.

5.2.2 Configuring Datasource

SpagoBI needs a JNDI datasource for the metadata database. To configure the JNDI resources do the following steps:

- Copy from **EXO-INST-FILES**/server/default/deploy/ the file spagobi-ds.xml into **EXO-HOME**/server/default/deploy/.
- Edit the spagobi-ds.xml file and change the properties `connection-url`, `driver-class`, `user-name`, `password`, `type-mapping` with the right value for the connection to your SpagoBI database. (The default value are suitable for hsqldb).

5.3 Install SpagoBI platform

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the 'SpagoBI-bin-1.8.zip'. Extract from the zip archive the file **SpagoBI/spagobi.war** and unzip it into **EXO-HOME**/server/default/deploy/exoplatform.sar/spagobi.war folder. (To unzip a war file you need first to rename its extension from war to zip). At the end you should have a 'spagobi.war' folder (into exoplatform.sar directory) which contains other subfolders 'jsp', 'img', 'WEB-INF', etc.
- Edit the file application.xml in **EXO-HOME**/server/default/deploy/exoplatform.sar/META-INF folder and add the following module (add only the <module> xml envelope into the existing <application> xml envelope):

```
<application>
...
<module>
  <web>
    <web-uri>spagobi.war</web-uri>
    <context-root>spagobi</context-root>
  </web>
</module>
...
</application>
```

- Copy the file **jboss-web.xml** from **EXO-INST-FILES**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF into **EXO-HOME**/server/default/deploy/exoplatform.sar /spagobi.war/WEB-INF. *Explanation: In JBoss we need to prepare a JBoss*

web deployment descriptor to introduce a level of indirection between the handling of JNDI resource and the name that this resource has in web.xml deployment descriptor. In that case with JBoss web deployment descriptor we're able to map a resource defined in web.xml to the one defined in JBoss space.

- Rename the file **EXO-INST-FILES**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/web.xml into web_origin.xml
- Copy the file **web.xml** from **EXO-INST-FILES**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF into **EXO-HOME**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF
- Delete the jar commons-logging-1.0.jar from **EXO-HOME**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/lib.

5.3.1 Configure Hibernate

SpagoBI uses Hibernate to manage metadata. Since SpagoBI can use different databases, hibernate must be configured in order to use the correct settings and sql dialect. Into the folder exo-home/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/classes there are four hibernate configuration files, one for each database supported:

- hibernate.cfg.xml (postgres database)
- hibernate.cfg.ora.xml (oracle database)
- hibernate.cfg.mysql.xml (mysql database)
- hibernate.cfg.hsql.xml (hsql database)

You must define which file must be used based on your database:

- edit the file exo-home/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/conf/spagobi/spagobi.xml
- search for the tag <HIBERNATE-CFGFILE>
- change its value putting the name of the correct hibernate configuration file (the default is for hsqldb)

5.4 CMS configuration

SpagoBI needs a connection to a content management system (cms) compliant to the jsr 170 specification in order to store and version the BI documents. The connection is represented by a jsr 170 'Repository' object which allows to open working session into the cms. SpagoBI can be configured to initialize directly the repository or to get it as a jndi resource. Using JBoss server it's necessary to initialize the repository directly, so:

- Only if you haven't already done it, copy the folder **EXO-INST-FILES**/sbidata into **EXO-HOME** folder.
- open the file jackrabbitSessionFactory.properties in **EXO-HOME**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/classes and substitute \${EXO_HOME} with the actual path of **EXO-HOME** in the property 'repository_path' (this property indicates the folder in which JackRabbit will store contents). Pay attention at the path form, also for Windows system it must contain only / separator and it must start with / (no c:\ for example); An example path could be '/Programs/exo-jboss'
- rename the file 'cms.xml' (contained into **EXO-HOME**/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/conf folder) into 'cms_origin.xml'

- rename the file 'cms-jackrabbit.xml' (contained into EXO-HOME/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/conf folder) into 'cms.xml'.
- At last you have to substitute the file commons-collection.jar in EXO-HOME/server/default/lib with the commons-collection-3.1.jar you find in EXO-HOME/server/default/deploy/ exoplatform.sar/spagobi.war/WEB-INF/lib

5.5 Engines

5.5.1 Install SpagoBIJasperReportEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJasperReportEngine-bin-1.8.zip'. Extract from the zip archive the file SpagoBI/SpagoBIJasperReportEngine.war and rename it as SpagoBIJasperReportEngine.zip. Create a folder with the name SpagoBIJasperReportEngine.war and unzip the file SpagoBIJasperReportEngine.zip inside it. Delete the SpagoBIJasperReportEngine.zip file and the copy the parent SpagoBIJasperReportEngine.war directory into **EXO-HOME**/server/default/deploy folder.
- Remove the following jar files from **EXO-HOME**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF/lib
 - commons-logging-api-1.0.2.jar
 - commons-logging-1.0.2.jar
 - xalan.jar
 - xercesImpl.jar
 - xml-apis.jar
 - log4j-1.2.8.jar
- Copy the file **jboss-web.xml** from **EXO-INST-FILES**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF into the folder **EXO-HOME**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF.
- Rename the file **EXO-INST-FILES**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF/web.xml into web-orig.xml
- Copy the file **web.xml** from **EXO-INST-FILES**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF folder into the folder **EXO-HOME**/server/default/deploy/SpagoBIJasperReportEngine.war/WEB-INF.

5.5.2 Install SpagoBIJPivotEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagBIJPivotEngine-bin-1.8.zip'. Extract from the zip archive the file SpagoBI/SpagoBIJPivotEngine.war and rename it as SpagoBIJPivotEngine.zip. Create a new folder named SpagoBIJPivotEngine.war and unzip the file SpagoBIJPivotEngine.zip inside it. Delete the SpagoBIJPivotEngine.zip file and copy the parent SpagoBIJPivot.war directory into **EXO-HOME**/ server/default/deploy folder.

- Copy the file **jboss-web.xml** from **EXO-INST-FILES**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF into the folder **EXO-HOME**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF.
- Rename the file **EXO-INST-FILES**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF/web.xml into web_origin.xml
- Copy the file **web.xml** from **EXO-INST-FILES**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF folder into the **EXO-HOME**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF folder.
- Remove the following jar files from **EXO-HOME**/server/default/deploy/SpagoBIJPivotEngine.war/WEB-INF
 - commons-logging-1.0.4.jar
 - log4j-1.2.8.jar

5.6 Install Example Portal

After the execution of the previous steps you should be able to connect to exo portal, imports SpagoBI portlets and define your own portal pages. (See exo documentation for information and instructions). However our distribution contains also a simple portal, which can be installed over exo platform, useful to test SpagoBI portlets. To install the test portal:

- copy the **EXAMPLE-PORTAL**/sbiportal.war into **EXO-HOME**/apps/autoload/exoplatform.ear folder
- Edit the file application.xml in **EXO-HOME**/apps/autoload/exoplatform.ear/META-INF folder and add the following module (add only the <module> xml envelope into the existing <application> xml envelope):

```
<application>
...
<module>
  <web>
    <web-uri>sbiportal.war</web-uri>
    <context-root>sbiportal</context-root>
  </web>
</module>
...
</application>
```

- start the server

At the end you should be able to connect to the url <http://localhost:9000/sbiportal>. The example portal defines four users:

- biadmin (password=biadmin): his pages contains the SpagoBI administration portlet
- bidev (password=bidev): his pages contains the SpagoBI development portlet
- bitest (password=bitest): his page contains the SpagoBI execution portlet with the possibility to view documents in test state
- biuser (password=biuser): his page contains the SpagoBI execution portlet