

SpagoBI JOnAS Installation Manual 1.4.3

Authors:

Fiscato Luca
Zerbetto Davide

Index

1 VERSION.....	3
2 DOCUMENT GOAL.....	3
3 REFERENCES.....	3
4 INSTALL SPAGOBI.....	4
4.1 INSTALL ADDITIONAL LIBRARIES AND CONFIGURATION FILES.....	4
4.2 INSTALL THE METADATA DATABASE.....	4
4.2.1 <i>Install the database driver.....</i>	5
4.2.2 <i>Configuring Datasource as a JNDI Resource.....</i>	5
4.3 INSTALL SPAGOBI CORE.....	6
4.3.1 <i>Configure Hibernate.....</i>	7
4.3.2 <i>Configure Quartz.....</i>	7
4.4 CMS CONFIGURATION.....	8
4.5 ENGINES.....	9
4.5.1 <i>Install SpagoBIJasperReportEngine.....</i>	9
4.5.2 <i>Install SpagoBIJPivotEngine</i>	9
4.5.3 <i>Install SpagoBIQbeEngine.....</i>	9
4.5.4 <i>Install SpagoBIBirtReportEngine</i>	10
4.5.5 <i>Install SpagoBIWekaEngine</i>	10
4.5.6 <i>Install SpagoBIGeoEngine</i>	10
4.5.7 <i>Install SpagoBIJPXMLAEngine</i>	11
4.6 INSTALL EXAMPLE PORTAL.....	12

1 Version

Version/Release n° :	1.2	Data Version/Release :	Jan, 31th 2006
Update description:	JOnAS Installation steps		
Version/Release n° :	1.3	Data Version/Release :	April, 7th 2006
Update description:	JOnAS 4.6.6 server, with eXo Portal 1.1, installation steps		
Version/Release n° :	1.4	Data Version/Release :	July, 14th 2006
Update description:	SpagoBI 1.9 installation steps on a JOnAS 4.6.6 server with eXo Portal 1.1.1.		
Version/Release n° :	1.4.1	Data Version/Release :	October, 9th 2006
Update description:	SpagoBI 1.9.1 installation steps on a JOnAS 4.7.4 server with eXo Portal 1.1.3.		
Version/Release n° :	1.4.2	Data Version/Release :	January, 23th 2007
Update description:	SpagoBI 1.9.2 installation steps on a JOnAS 4.7.4 server with eXo Portal 1.1.3.		
Version/Release n° :	1.4.3	Data Version/Release :	February, 15th 2007
Update description:	Added SpagoBIJPXMLAEngine Installation		

2 Document goal

This document provides a step by step description for SpagoBI installation and configuration on a JOnAS 4.7.4 server with eXo Portal 1.1.3 installed. To get a working copy of JOnAS 4.7.4 server with eXo Portal 1.1.3 you have to:

- download from JOnAS ObjectWeb site (<http://forge.objectweb.org/projects/jonas>) the file jonas4.7.4-tomcat5.5.15.tgz (you can find it selecting the 'files' tab) and unzip it into your file system. (JONAS-HOME)
- download, from eXo ObjectWeb site (<http://forge.objectweb.org/projects/exoplatform>), the file eXoPortal-JOnAS-1.1.3.zip (you can find it selecting the 'files' tab) and unzip it into your file system (EXO-JONAS)
- follow the eXo installation instructions that you find in the Readme file contained into eXoPortal-JOnAS-1.1.3.zip.

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>)
- JOnAS application server (available at <http://jonas.objectweb.org>)

4 Install SpagoBI

We assume that you have correctly installed eXo Portal 1.1.3 on a JOnAS 4.7.4 server; in the following we will refer to the exo-JOnAS 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 'Files' tab, download the file called 'SpagoBIUtilityFiles-bin-1.9.2.zip', and then unzip it.

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

4.1 Install additional libraries and configuration files

SpagoBI needs some additional libraries and configuration files in order to start correctly and manage metadata.

1. Copy the content of **EXO-INST-FILES/lib/apps** inside your **EXO-HOME/lib/apps** directory (*).
2. Delete the file **EXO-HOME/lib/apps/hsqldb-1.8.0.1.jar** (the previous operation copy a more recent version of hsqldb and so the old one has to be deleted)
3. Copy the **EXO-INST-FILES/sbidata** folder into your **EXO-HOME**. The folder contains the hsqldb metadata database and the root folder of the cms repository.

(*) The following libraries will be copied into **EXO-HOME/lib/apps**:

- ehcache-1.1.jar (hiberante cache)
- concurrent-1.3.2.jar (jackrabbit dependency)
- hsqldb1_8_0_2.jar (more recent version of the hsqldb)
- jackrabbit-core-1.0.1.jar (for the cms repository)
- jcr-1.0.jar (jackrabbit dependency)
- slf4j-log4j12.jar (jackrabbit dependency)

4.2 Install the Metadata Database

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

SpagoBI Utility files package contains an hsql database that can be used to test SpagoBI without installing a database server. After the execution of the activities listed in the previous paragraph the hsqldb script is stored into **EXO-HOME/sbidata/database**.

HSQL database is very useful for test purpose but is very weak in a production environment. Anyway, if you want to use HSQLDB, just 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' databases already populated with the necessary data. **Remember that every time you start the exo server to work with SpagoBI the database server must be running.**

If you don't want to use hsqldb you have the possibility to choose between PostgreSQL, Oracle and MySQL. If you don't have anyone of these database servers installed 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. So, connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file called <<name of you database>>-dbscript-1.9.2.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-1.9.2.zip) and run in the following order the scripts PG_create.sql and PG_insert.sql*

4.2.1 Install the database driver

Based on the database server you chose for the SpagoBI metadata you must install the right database driver. The drivers package can be obtained from database vendor sites 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/lib/apps)

The driver jar, corresponding to the database you use, have to be put under **EXO-HOME/lib/commons/jonas**. If you decide to use the SpagoBI hsqldb database you have to add into **EXO-HOME/lib/commons/jonas** the hsqldb1_8_0_2.jar library and to remove the existing hsqldb.jar library in the same folder in order to avoid conflicts.

4.2.2 Configuring Datasource as a JNDI Resource

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

- Copy from EXO-INST-FILES/conf the files spagobi.properties into EXO-HOME/conf
- Edit the file spagobi.properties. The default values are suitable for the SpagoBI hsql test database. If you don't use the hsql database you need to fill the value of the following properties with the right value for the connection to your SpagoBI database:
 - datasource.url
 - datasource.classname
 - datasource.username
 - datasource.password
 - datasource.mapper
- Edit the file EXO-HOME/conf/jonas.properties, search the property jonas.service.dbm.datasources and add to its value the string “,spagobi”. As an example if the original row is 'jonas.service.dbm.datasources HSQL1' it must become 'jonas.service.dbm.datasources HSQL1,spagobi'.

4.3 Install SpagoBI core

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the 'SpagoBI-bin-1.9.2.zip'. Extract from the zip archive the file spagobi.war and unzip it into EXO-HOME/apps/autoload/exoplatform.ear/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.ear directory) which contains other subfolders 'jsp', 'img', 'WEB-INF', etc.
- 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>spagobi.war</web-uri>
      <context-root>spagobi</context-root>
    </web>
  </module>
...
</application>
```

- Delete the jar commons-logging-1.0.jar from EXO-HOME/apps/autoload/exoplatform.ear /spagobi.war/WEB-INF/lib.
- Delete the jar log4j-1.2.8.jar from EXO-HOME/apps/autoload/exoplatform.ear /spagobi.war/WEB-INF/lib.
- Edit the file spagobi.xml contained into EXO-HOME/apps/autoload/exoplatform.ear/spagobi.war/WEB-INF/conf and:

- Search the tag <SPAGOBI_CONTEXT_PATH> and replace the value with your SpagoBI installation URL (you have simply to change the server name and port). Example: if you have installed SpagoBI on a server call myhost and the server is listening on port 5000 the value must be <http://myhost:5000/spagobi>.
- Search the tag <LANGUAGE_SUPPORTED> and set to true the 'default' attribute of the children LANGUAGE tag, corresponding to your language. Be sure that one and only one of the LANGUAGE tags have the 'default' attribute set to true.

4.3.1 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/apps/autoload/exoplatform.ear/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)

Since JOnAS has a different jndi tree management system it's necessary to:

- edit the hibernate configuration file suitable for your database
- search the tag '<property name="hibernate.connection.datasource">'
- replace its original value ('java:comp/env/jdbc/spagobi') with the new value 'jdbc/spagobi'

Then you must define which file must be used, based on your database:

- edit the file `exo-home/apps/autoload/exoplatform.ear/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)

4.3.2 Configure Quartz

The SpagoBI scheduler feature is based on Quartz technology. Quartz is a library which can store its metadata into a database. For the SpagoBI installation the quartz metadata database can be the same as the SpagoBI one. In order to configure it do the following steps:

- edit the file `EXO-HOME/apps/autoload/exoplatform.ear/spagobi.war/WEB-INF/classes/quartz.properties`
- search the string `org.quartz.dataSource.quartz.jndiURL` and change its value to 'jdbc/spagobi' (instead of 'java:comp/env/jdbc/spagobi')
- search the string 'job store delegate class'. Under this split line there's the same property repeated four times (`org.quartz.jobStore.driverDelegateClass`)
- based on your SpagoBI database you have to uncomment the right one and obviously to comment the others (to comment a row just place a # at the beginning)

4.4CMS 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 JOnAS 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 cms-jboss-jonas.xml in EXO-HOME/apps/autoload/exoplatform.ear/spagobi.war/WEB-INF/conf and substitute \${SERVER_HOME} with the actual path of EXO-HOME in the definition of the parameters 'repository_path' (this property indicates the folder in which JackRabbit will store contents) and 'conf_file_path' (this property indicates the file for Jackrabbit configuration). 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'
- delete the file cms.xml (contained into EXO-HOME/apps/autoload/exoplatform.ear/spagobi.war/WEB-INF/conf folder) or rename it into 'cms_tomcat.xml'
- rename the file cms-jboss-jonas.xml (contained into EXO-HOME/server/default/deploy/exoplatform.sar/spagobi.war/WEB-INF/conf folder) into cms.xml.

4.5Engines

4.5.1Install SpagoBIJasperReportEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJasperReportEngine-bin-1.9.2.zip'. Extract from the zip archive the file SpagoBIJasperReportEngine.war. Unzip the war file into a directory named SpagoBIJasperReportEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: Report
 - engine type: External
 - driver class: it.eng.spagobi.drivers.jasperreport.JasperReportDriver
 - url: `http://<server>:<port>/SpagoBIJasperReportEngine/JasperReportServlet`.

4.5.2Install SpagoBIJPivotEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJPivotEngine-bin-1.9.2.zip'. Extract from the zip archive the file SpagoBIJPivotEngine.war. Unzip the war file into a directory named SpagoBIJPivotEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: On-line analytical processing
 - engine type: External
 - driver class: it.eng.spagobi.drivers.jpivot.JPivotDriver
 - url: `http://<server>:<port>/ SpagoBIJPivotEngine/jpivotOlap.jsp`.

4.5.3Install SpagoBIQbeEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIQbeEngine-bin-1.9.2.zip'. Extract from the zip archive the file SpagoBIQbeEngine.war. Unzip the war file into a directory named SpagoBIQbeEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- If you want (advised) change the password of the administrator user (look at the section 'How to configure SpagoBIQbeEngine administrator users' of the How To documentation file to learn how to do it)
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: Datamart Model
 - engine type: External
 - driver class: it.eng.spagobi.drivers.qbe.QbeDriver

○url: `http://<server>:<port>/SpagoBIQbeEngine/servlet/AdapterHTTP?ACTION_NAME=SPAGO_BI_START_ACTION&NEW_SESSION=TRUE` (without any space in the middle)

4.5.4 Install SpagoBIBirtReportEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagBIBirtReportEngine-bin-1.9.2.zip'. Extract from the zip archive the file SpagoBIBirtReportEngine.war. Unzip the war file into a directory named SpagoBIBirtReportEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: Report
 - engine type: External
 - driver class: `it.eng.spagobi.drivers.birt.BirtReportDriver`
 - url: `http://<server>:<port>/SpagoBIBirtReportEngine/BirtReportServlet`.

4.5.5 Install SpagoBIWekaEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagBIWekaEngine-bin-1.9.2-Alpha'. Extract from the zip archive the file SpagoBIWekaEngine.war. Unzip the war file into a directory named SpagoBIWekaEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: Data mining model
 - engine type: External
 - driver class: `it.eng.spagobi.drivers.weka.WekaDriver`
 - url: `http://<server>:<port>/SpagoBIWekaEngine/WekaServlet`

4.5.6 Install SpagoBIGeoEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagBIGeoEngine-bin-1.9.2-Alpha'. Extract from the zip archive the file SpagoBIGeoEngine.war. Unzip the war file into a directory named SpagoBIGeoEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: Map
 - engine type: External
 - driver class: `it.eng.spagobi.drivers.geo.GeoDriver`
 - url: `http://<server>:<port>/SpagoBIGeoEngine/servlet/AdapterHTTP?ACTION_NAME=GEO_ACTION&NEW_SESSION=TRUE`. (without any space)

4.5.7 Install SpagoBIJPXMLAEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJPXMLAEngine-bin-1.9.2.zip'. Extract from the zip archive the file SpagoBIJPXMLAEngine.war. Unzip the war file into a directory named SpagoBIJPXMLAEngine.war. Copy the new directory into EXO-HOME/webapps/autoload folder.
- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJPXMLADriver-bin-1.9.2.zip'. Extract from the zip archive the file sbi.drivers.jpxmla-1.9.2.jar and copy it inside EXO-HOME/apps/autoload/exoplatform.ear/spagobi.war/WEB-INF/lib.
- Using the SpagoBI administration portlet define a new External Engine with:
 - document type: On-line analytical processing
 - engine type: External
 - driver class: it.eng.spagobi.drivers.jpxmla.JPXMLADriver
 - url: `http://<server>:<port>/SpagoBIJPXMLAEngine/JPivotServlet`

4.6 Install Example Portal

After the execution of the previous steps you should be able to connect to exo portal, import SpagoBI portlets and define your own portal pages. (Look at 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/jonas/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>
```

- copy the file 'sbiportal.script' and 'sbiportal.properties' from EXAMPLE-PORTAL/ jonas/db to EXO-HOME/temp/data (if the folder does not exist, create it);
- 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