SpagoBI eXo-WebOS Tomcat Installation Manual 1.0.0

Authors

Fiscato Luca
Zerbetto Davide

Index

1 Version

Version/Release n° :	1.0.0	**Data Version/Release :**	October, 30th 2007
Update description:	SpagoBI 1.9.4 installation steps on a Tomcat Server 6.0.x with eXo Enterprise WebOS		

2 Document goal

This document provides a step by step description for SpagoBI 1.9.4 installation and configuration on a Tomcat Server 6.0.x with eXo Enterprise WebOS installed. To get a working copy of Tomcat Server 6.0.x with eXo Enterprise WebOS you can download, from eXo OW2 Forge site (http://forge.objectweb.org/projects/exoplatform), the file exo-enterprise-webos-rxxxxx-tomcat.zip and simply unzip it into your file system. As an alternative you can download eXo platform source code and build the Portal WebOS product only: see eXo documentation to see how to compile and build eXo products.

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)
- Tomcat server (available at http://tomcat.apache.org/)

4 Install SpagoBI

We assume that you use the **java version 1.5.x** and that you have correctly installed the eXo Enterprise WebOS on a Tomcat 6.0.x server; in the following we will refer to the exo-tomcat base directory as **EXO-HOME**.

To test the eXo installation go to **EXO-HOME/bin** and run the command **eXo.bat run** (in a windows environment) or **eXo.sh run** (in a unix environment), wait until the command ends up with the message 'Server startup in xxxx ms' and then connect with a browser to the url http://<<localhost>>:8080/portal, the eXo home page should appear. Before proceed stop the server.

Connect to the SpagoBI page on the OW2 Forge (http://forge.objectweb.org/projects/spagobi), click on the 'File' tab, download the file called 'SpagoBIUtilityFiles-1.9.4_xxx.zip', and then unzip it.

The unzip operation produces a folder 'SpagoBIUtilityFiles' which contains 'tomcat-6.0.x', 'tomcat-server', 'jboss-server' and other folders. The folder 'tomcat-6.0.x' respects the tree-folders structure of a Tomcat 6.0.x and contains some required libraries and files; in the following we will refer to this folder as **EXO-INST-FILES**.

4.1Install 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 inside your **EXO-HOME**/lib directory (*).
2. 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**/common/lib:

- ehcache-1.1.jar (hibernate cache)
- jackrabbit-core-1.0.1.jar (for the cms repository)
- slf4j-log4j12.jar (jackrabbit dependecy)
- xercesImpl-2.6.2.jar (spago dependency)

4.2Install the Metadata Database

SpagoBI metadata are stored in a database (for this release SpagoBI supports PostgresSQL, Oracle, MySQL, SQLServer 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 with a 'spagobi' databases already populated with the necessary data. **Remember that every time you start eXo-Tomcat 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 PostgresSQL, Oracle SQLServer 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-1.9.4.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-1.9.4.zip) and run in*

the following order the scripts PG_create.sql, PG_insert.sql and then also PG_create_quartz_schema.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 or later versions
- SQLServer: sqljdbc.jar

The driver jar, corresponding to the database you use, have to be put under **EXO-HOME**/lib. If you decide to use the SpagoBI hsqldb database you have to do nothing because **EXO-HOME**/lib already contains a suitable driver (hsqldb-1.8.0.7.jar).

4.2.2 Configuring Datasource as JNDI Resource

SpagoBI needs a connection to the metadata database. This connection can be configured as a direct jdbc connection or as link to a server jndi datasource. The second option is surely better and it’s covered into this manual. To configure the JNDI global datasource resource do the following steps:

- Edit the file EXO-INST-FILES/conf/server.xml.
- Search for the string “<Resource name="jdbc/spagobi"”;
- Copy entirely the <Resource name="jdbc/spagobi" .... >” tag
- Edit the file EXO-HOME/conf/server.xml
- Inside tag named ‘GlobalNamingResources’ paste the piece of xml copied previously

The xml copied configures a new jndi datasource for the metadata database. The default values are for the SpagoBI hsql database and, if you are using another database server, you need to change them. Based on your database change the value of the following parameters:

- driverClassName
- url
- username
- password

4.3 Install Cms Repository

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. The second option is surely better and it's covered into this manual. To configure the JNDI global repository resource do the following steps:

- Edit the file EXO-INST-FILES/conf/server.xml.
- Search for the string "<Resource name="cms/spagobicms"";
- Copy entirely the <Resource name="cms/spagobicms" ... >" tag
- Edit the file EXO-HOME/conf/server.xml
- Inside tag named 'GlobalNamingResources' paste the piece of xml previously copied

The xml copied contains two parameters which have to be configured

- configFilePath (path of the jackrabbit configuration file)
- repHomeDir (path to the folder that will be the root of the repository)

You have simply to replace the string '${SERVER_HOME}' with the path of your eXo tomcat installation directory. Example: if your eXo tomcat is installed into
C:\Programs\exo-tomcat
you have to replace '${SERVER_HOME}' with '/Programs/exo-tomcat'

4.4 Install SpagoBI platform

Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the 'SpagoBI-bin-1.9.4_xxx.zip'. Extract from the zip archive the file spagobi.war and rename it as spagobi.zip. Create a new folder named "spagobi" and unzip the file spagobi.war file inside it. Delete the spagobi.zip file and copy the parent "spagobi" directory into **EXO-HOME**/webapps folder.

Remove the following jar files from **EXO-HOME**/webapps/spagobi/WEB-INF/lib:

1. commons-codec-1.3.jar
2. commons-collections-3.1.jar
3. commons-dbcp-1.2.1.jar
4. commons-httpclient-3.0.1.jar
5. commons-javaflow-20060411.jar
6. commons-logging-1.0.jar
7. commons-pool-1.2.jar
8. log4j-1.2.8.jar

Copy the file spagobi.xml from **EXO-INST-FILES**/conf/Catalina/localhost to **EXO-HOME**/conf/Catalina/localhost.

Edit the file spagobi.xml contained in EXO-HOME/webapps/spagobi/WEB-INF/conf/spagobi 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 (default Tomcat server port is 8080).
- 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.

4.4.1Configure 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/webapps/spagobi/WEB-INF/classes there are five hibernate configuration files, one for each database supported:

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

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

- edit the file exo-home/webapps/spagobi/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)

4.4.2 Configure Quartz

The SpagoBI scheduler feature is based on Quartz technology. Quartz is a library which can store its metadata into a database: in the default configuration this database is the SpagoBI one. In order to properly configure it do the following steps:

- edit the file EXO-HOME/webapps/spagobi/WEB-INF/classes/quartz.properties
- 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 server (HSQLDB, Oracle, ...) you have to uncomment the right one and obviously to comment the others (to comment a row just place a # at the beginning)

4.5 Install Engines

4.5.1 Install SpagoBIJasperReportEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIJasperReport-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJasperReportEngine.war and copy it into the EXO_HOME/webapps folder;
- copy the file SpagoBIJasperReportEngine.xml from EXO-INST-FILES/conf/Catalina/localhost into EXO-HOME/conf/Catalina/localhost.
- 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.

4.5.2 Install SpagoBIJPivotEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIJPivotEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJPivotEngine.war and copy it into the EXO-HOME/webapps folder.
- Copy the file SpagoBIJPivotEngine.xml from EXO-INST-FILES/conf/Catalina/localhost into EXO-HOME/conf/Catalina/localhost
- 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/JPivotServlet.

4.5.3 Install SpagoBIQbeEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIQbeEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIQbeEngine.war and copy it into the EXO-HOME/webapps 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)
- Remove the following jar file from EXO-HOME/webapps/SpagoBIQbeEngine/WEB-INF/lib
    1. commons-logging-1.0.4.jar
- 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)

4.5.4 Install SpagoBIBirtReportEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIBirtReportEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIBirtReportEngine.war and copy it into the EXO-HOME/webapps folder.
- Copy the file SpagoBIBirtReportEngine.xml from EXO-INST-FILES/conf/Catalina/localhost into EXO-HOME/conf/Catalina/localhost
- 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 'SpagoBIWekaEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIWekaEngine.war and copy it into the EXO-HOME/webapps folder.
- Copy the file SpagoBIWekaEngine.xml from EXO-INST-FILES/conf/Catalina/localhost into EXO-HOME/conf/Catalina/localhost
- 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 'SpagoBIGeoEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIGeoEngine.war and copy it into the EXO-HOME/webapps folder.
- Copy the file SpagoBIGeoEngine.xml from EXO-INST-FILES/conf/Catalina/localhost into EXO-HOME/conf/Catalina/localhost
- 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

4.5.7 Install SpagoBIJPXMLAEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIJPXMLAEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJPXMLAEngine.war and copy it into the EXO-HOME/webapps folder.
- The driver is the same of SpagoBIJPivotEngine: control that the file sbi.driver.jpivot-1.9.4.jar is present inside folder EXO-HOME/webapps/spagobi/WEB-INF/lib, if it is missing you have to connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIJPivotDriver-bin-1.9.4_xxx.zip'. Extract from the zip archive the file sbi.drivers.jpivot-1.9.4.jar and copy it inside EXO-HOME/webapps/spagobi/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.jpivot.JPivotDriver
    - ○url: http://<server>:<port>/SpagoBIJPXMLAEngine/JPivotServlet

Note that SpagoBIJPXMLAEngine is a client web application for a XMLA server. The installation steps of a XMLA server are not included in this document.

4.5.8Install SpagoBITalendEngine

- Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBITalendEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBITalendEngine.war and rename it as SpagoBITalendEngine.zip. Create a new folder named SpagoBITalendEngine and unzip the file SpagoBITalendEngine.zip inside it. Delete the SpagoBITalendEngine.zip file and copy the new SpagoBITalendEngine directory into EXO-HOME/webapps folder.
- Edit file talend.properties in EXO-HOME/webapps/SpagoBITalendEngine/WEB-INF/classes and configure the following properties:
    - `runtimeRepository.rootDir`: the root path of the repository containing the jobs: it can be absolute or relative(*);
    - `spagobi.autopublish`: if it is true, when you deploy a new job from Talend Opend Studio, a SpagoBI document will be automatically created and put on the functionalities tree;
    - `spagobi.functionality.label`: the label of the functionality where SpagoBI document will be created (if `spagobi.autopublish` is true);
    - `spagobi.url`: the SpagoBI context url: it is required if `spagobi.autopublish` is true.
- Edit file talend.perl.properties in EXO-HOME/webapps/SpagoBITalendEngine/WEB-INF/classes and adjust your Perl installation directory (mandatory if you want to executed Perl based jobs)(*).
- Edit file talend.java.properties in EXO-HOME/webapps/SpagoBITalendEngine/WEB-INF/classes and adjust Java process memory options; if you use a unix based environment, you have to adjust also your Java installation directory(*).
- Using the SpagoBI administration portlet define a new External Engine with:
    1. document type: ETL

2.engine type: External
3.driver class: it.eng.spagobi.drivers.talend.TalendDriver
4.url:http://<server>:<port>/SpagoBITalendEngine/JobRunService.

(*) Pay attention at the path form, also for Windows system it must contain only / separator and it must start with / (no c:\ for example).