SpagoBI 1.9.4 Web Application for Jboss 4.0.3SP1 Installation Manual 1.0.0a

Authors

Yannick LE NY

from the original document
SpagoBI Web Application Tomcat Installation Manual 1.0.0
created by
Bernabei Angelo
Zerbetto Davide

Index

1 Version

Version/Release n° :	1.0.0	**Data Version/Release :**	September, 26th 2008
Update description:	SpagoBI 1.9.4 Web Application installation steps on a Jboss Server 4.0.3SP1 by Yannick LE NY from the original document "SpagoBI Web Application Tomcat Installation Manual 1.0.0" created by Bernabei Angelo and Zerbetto Davide		
Version/Release n° :	1.0.0a	**Data Version/Release :**	October, 2th 2008
Update description:	Minor corrections by Yannick LE NY		

2 Document goal

This document provides a step by step description for SpagoBI 1.9.4 Web Application installation and configuration on a Jboss Server 4.0.3SP1. To get a working copy of a JBoss 4.0.3SP1 server you can download, from Jboss download page, the file jboss-4.0.3SP1.zip and simply unzip it into your file system:

- download, from the JBoss site, the version JBoss 4.0.3SP1 and unzip it into your file system (jboss-home).
  Here there is the direct link to the jboss-4.0.3SP1.zip (MS Windows) or jboss-4.0.3SP1.tar.gz (Linux/Unix) file:
  `http://sourceforge.net/project/showfiles.php?group_id=22866&package_id=16942&release_id=365509`
- remove all the libraries contained into jboss-home/lib/endorsed (resolver.jar, xalan.jar, xercesImpl.jar, xml-apis.jar)

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/)
- Spago framework (available at http://spago.eng.it)
- JBoss application server (available at http://www.jboss.org)

4 Install SpagoBI

We assume that you use the **java version 1.5.x** and that you have correctly installed a Jboss 4.0.3SP1 server; in the following we will refer to the Jboss base directory as **JBOSS-HOME** or here the /home/spagobi/spagobi_194/jboss-4.0.3SP1 directory. .
To test the Jboss installation go to **JBOSS-HOME/bin** and run the command **run.bat** (in a windows environment) or **run.sh** (in a unix environment), wait until the command ends up with the message '[Server] JBoss (MX MicroKernel) [4.0.3SP1 (build: CVSTag=JBoss_4_0_3_SP1 date=200510231054)] Started in xs:xxx ms' and then connect with a browser to the url

http://<<localhost>>:8080/, the Tomcat default home page should appear. Before proceed stop the server.

1) We choose to install Jboss in the /home/spagobi/spagobi_194
2) Copy the file jboss-4.0.3SP1.tar.gz in the /home/spagobi/spagobi_194 directory or a directory that you choose for installing.
3) Go in the /home/spagobi/spagobi_194 directory and uncompress the jboss-4.0.3SP1.tar.gz file with tar ang gzip command line.
4) remove all the libraries contained into /home/spagobi/spagobi_194/jboss-4.0.3SP1/lib/endorsed (resolver.jar, xalan.jar, xercesImpl.jar, xml-apis.jar)
5) Here we have now the /home/spagobi/spagobi_194/jboss-4.0.3SP1 directory for the jboss-home directory.

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 'spagobi-web', 'tomcat-server', 'jboss-server' and other folders.
The folder 'jboss-server' respects the tree-folders structure of a Jboss4.0.3SP1 installation nd contains some required libraries and files; in the following we will refer to this folder as **JBOSS-FILES**.
The folder 'spagobi-web' 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 **TOMCAT-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**.

4.1Install additional libraries and configuration files

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

1.. Copy the jar files contained in **JBOSS-INST-FILES**/server/default/deploy/exoplatform.sar inside your **JBOSS-HOME**/server/default/lib directory (*).
2.Copy the jar files contained in **TOMCAT-INST-FILES**/lib inside your **JBOSS-HOME**/server/default/lib (**).
3.Copy the content of **JBOSS-INST-FILES/lib** inside your **JBOSS-HOME**/server/default/lib (***).
4.. Copy the **JBOSS-INST-FILES**/sbidata folder into your **JBOSS-HOME.** The folder contains the hsqldb metadata database and the root folder of the cms repository.

(*) The following libraries will be copied into **JBOSS-HOME**/server/default/lib directory:

- ehcache-1.1.jar
- concurrent-1.3.4.jar

- jackrabbit-core-1.0.1.jar
- slf4j-log4j12.jar
- jcr-1.0.jar

(**) The following libraries will be copied into **JBOSS-HOME**/server/default/lib directory:

- portlet-api-1.0.jar
- dom4j-1.6.jar
- jta-1.0.1B.jar
- ldap.jar
- xercesImpl.jar

(***) The following libraries will be copied into **JBOSS-HOME**/server/default/lib directory:

- commons-collections-3.1.jar
- hsqldb1_8_0_2.jar (!! remember to remove the existing hsqldb.jar library in the same folder in order to avoid conflicts).

5. With Jboss 4.0.3SP1, you need jdt-compiler-3.0.2.jar file to use SpagoBI 1.9.4 as Web Application, but the file jdt-compiler-3.0.2.jar is not in the the file called 'SpagoBIUtilityFiles-1.9.4_xxx.zip'. You need to do some steps to install it.
a) Download the file eclipse-SDK-3.0.2-win32.zip
at http://archive.eclipse.org/eclipse/downloads/drops/R-3.0.2-200503110845/index.php
b) unzip the eclipse zip
c) go in the eclipse-3.0.2/eclipse/plugins/org.eclipse.jdt.core_3.0.2 directory
d) copy the jdtcore.jar file into **JBOSS-HOME**/server/default/lib directory
e) go into **JBOSS-HOME**/server/default/lib directory et rename the file jdtcore.jar to jdt-compiler-3.0.2.jar
Note : use only jdt-compiler-3.0.x.jar , jdt-compiler-3.1.x.jar don't work with Tomcat 5.5.X (5.5.9) include in Jboss 4.0.3SP1. (See https://www.zarb.org/pipermail/jpackage-discuss/2005-October/008945.html)

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 **JBOSS-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 **JBOSS-HOME**/sbidata/database/start.bat (on a windows platform) or **JBOSS-HOME**/sbidata/database/start.sh (on a linux/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 Jboss 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. (**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-1.9.4.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.*

4.2.1Install 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 JBOSS-INST-FILES/server/default/lib)
- SQLServer: sqljdbc.jar

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

Pay attention: If you decide to use the SpagoBI hsqldb example database add into **JBOSS-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.

4.2.2Configuring Datasource as JNDI Resource

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

- Copy from JBOSS-INST-FILES/server/default/deploy/ the file spagobi-ds.xml into JBOSS-HOME/server/default/deploy/.

  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

- 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).
- SpagoBI is provided with a hsqldb database for testing but not for production. (Host : localhost, port : 9002, user: sa , password : nothing, database : spagobi). This testing database is stored in the sbidata/database directory.

In the default spago-ds.xml , for testing with hsqldb database, you have this configuration :

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>spagobi</jndi-name>
    <connection-url>jdbc:hsqldb:hsql://localhost:9002/spagobi</connection-url>
    <driver-class>org.hsqldb.jdbcDriver</driver-class>
    <user-name>sa</user-name>
    <password></password>


    <min-pool-size>5</min-pool-size>

     <!-- The maximum connections in a pool/sub-pool -->
    <max-pool-size>20</max-pool-size>

    <metadata>
```

```
        <type-mapping>Hypersonic SQL</type-mapping>
      </metadata>

    </local-tx-datasource>
</datasources>
```

4.3 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 folder with the name spagobi.war and unzip the file spagobi.zip inside it. Delete the spagobi.zip file and then copy the parent spagobi.war directory into JBOSS-HOME/server/default/deploy/ folder. (To unzip a war file you need first to rename its extension form war to zip). At the end you should have a 'spagobi.war' folder which contains other subfolders 'jsp', 'img', 'WEB-INF', etc.

Edit the file spagobi.xml contained in JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/spagobi and:

- Search the tag <SPAGOBI-MODE> and change the attribute 'mode' to 'WEB':
  <SPAGOBI-MODE mode="WEB" />
- 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.
  Example: if you want to have menus and messages in french, you need this configuration :
  <LANGUAGE_SUPPORTED>
  <LANGUAGE default="false" language="it" country="IT" />
  <LANGUAGE default="**false**" language="en" country="US" />
  <LANGUAGE default="**true**" language="fr" country="FR" />.

  Note : Some menus and messages could be not translated because there is no translation file created for them.

- Edit the file initializer.xml contained into JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf and uncomment the following initializers:
  <INITIALIZER class="it.eng.spagobi.security.init.SecurityInitializer" config="" />
  <INITIALIZER class="it.eng.spagobi.init.TreeInitializer" config="SPAGOBI.TREE_INITIALIZATION" />
- Edit the file web.xml contained into JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF and comment the definitions and mappings of the portlet listener and portlet servlet:

```
[....]
<!--
<listener>
```

```
    <listener-class>org.exoplatform.services.portletcontainer.impl.servlet.PortletApplicationListener</listener-
class>
</listener>
-->
[....]
<!--
<servlet>
   <servlet-name>PortletWrapper</servlet-name>
   <servlet-class>org.exoplatform.services.portletcontainer.impl.servlet.ServletWrapper</servlet-class>
</servlet>
-->
[....]
<!--
<servlet-mapping>
    <servlet-name>PortletWrapper</servlet-name>
    <url-pattern>/PortletWrapper</url-pattern>
</servlet-mapping>
-->
[....]
```

- copy the file JBOSS-HOME/server/default/deploy/spagobi.war/components/mapcatalogue/img/mapManagement.png into JBOSS-HOME/server/default/deploy/spagobi.war/img/wapp directory.
  In the BOSS-HOME/server/default/deploy/spagobi.war/img/wapp directory, rename the file mapManagement.png to map64.png

4.3.1Configure 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 JBOSS-HOME/server/default/deploy/spagobi.war/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.hsql.xml (hsql database)
- hibernate.cfg.sqlserver.xml (sqlserver database)

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

- edit the file JBOSS-HOME/server/default/deploy/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)
- Example : the line for hsqldb databases is :

<HIBERNATE-CFGFILE>hibernate.cfg.hsql.xml</HIBERNATE-CFGFILE>

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: in the default configuration this database is the SpagoBI one. In order to properly configure it do the following steps:

- edit the file TOMCAT-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)

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 JBOSS-HOME/server/default/deploy/spagobi.war/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)
  - search the string org.quartz.dataSource.quartz.jndiURL and comment it (type a # as the first characther of the row)
  - search the string org.quartz.dataSource.quartz.driver and change its value putting the driver class name for your SpagoBI metadata database
  - search the string org.quartz.dataSource.quartz.URL and change its value putting the url of your SpagoBI metadata database
  - search the string org.quartz.dataSource.quartz.user and org.quartz.dataSource.quartz.password and change their values putting the username and password for the connection to your SpagoBI metadata database

- Example for testing with the hsqldb database provided with SpagoBI :

#---------- START JOB STORE ----------
#org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
org.quartz.jobStore.class = org.quartz.impl.jdbcjobstore.JobStoreTX
org.quartz.dataSource.quartz.jndiURL=java:comp/env/jdbc/spagobi

#org.quartz.dataSource.quartz.driver = **org.hsqldb.jdbcDriver**
#org.quartz.dataSource.quartz.URL = **jdbc:hsqldb:hsql://localhost:9002/spagobi**
#org.quartz.dataSource.quartz.user = **sa**
#org.quartz.dataSource.quartz.password =
#org.quartz.dataSource.quartz.maxConnections = **20**

#org.quartz.dataSource.quartz.driver = org.postgresql.Driver

#org.quartz.dataSource.quartz.URL = jdbc:postgresql://localhost:5432/quartz

#org.quartz.dataSource.quartz.user = postgres

#org.quartz.dataSource.quartz.password = postgres

#org.quartz.dataSource.quartz.maxConnections = 5

#---------- job store delegate class ----------
Hsqldb delegate class
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.HSQLDBDelegate

Mysql delegate class
#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.StdJDBCDelegate

Postgres delegate class
#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate

Oracle delegate class
#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
#----------

4.4 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. Using JBoss server it's necessary to initialize the repository directly, so:

- Only if you haven't already done it, copy the folder JBOSS-INST-FILES/sbidata into JBOSS-HOME folder.
- open the file cms-jboss-jonas.xml in JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf and substitute ${SERVER_HOME} with the actual path of JBOSS-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 JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf folder) or rename it into 'cms_tomcat.xml'
- rename the file cms-jboss-jonas.xml (contained into JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf folder) into cms.xml
- At last you have to substitute the file commons-collections.jar in JBOSS-HOME/server/default/lib with the commons-collections-3.1.jar you find in JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/lib

4.5 Authentication and authorization

Authentication and authentication can be perfomed connecting to an external Ldap system or using predefined xml files.

4.5.1Usign Ldap system

The provided implementation is based on Novel library (Open Source) to connect to OpenLDAP. When you have installed and configured OpenLDAP, you have to add group and user directories using its graphical client (JXPlorer) as per the following .ldif schema:

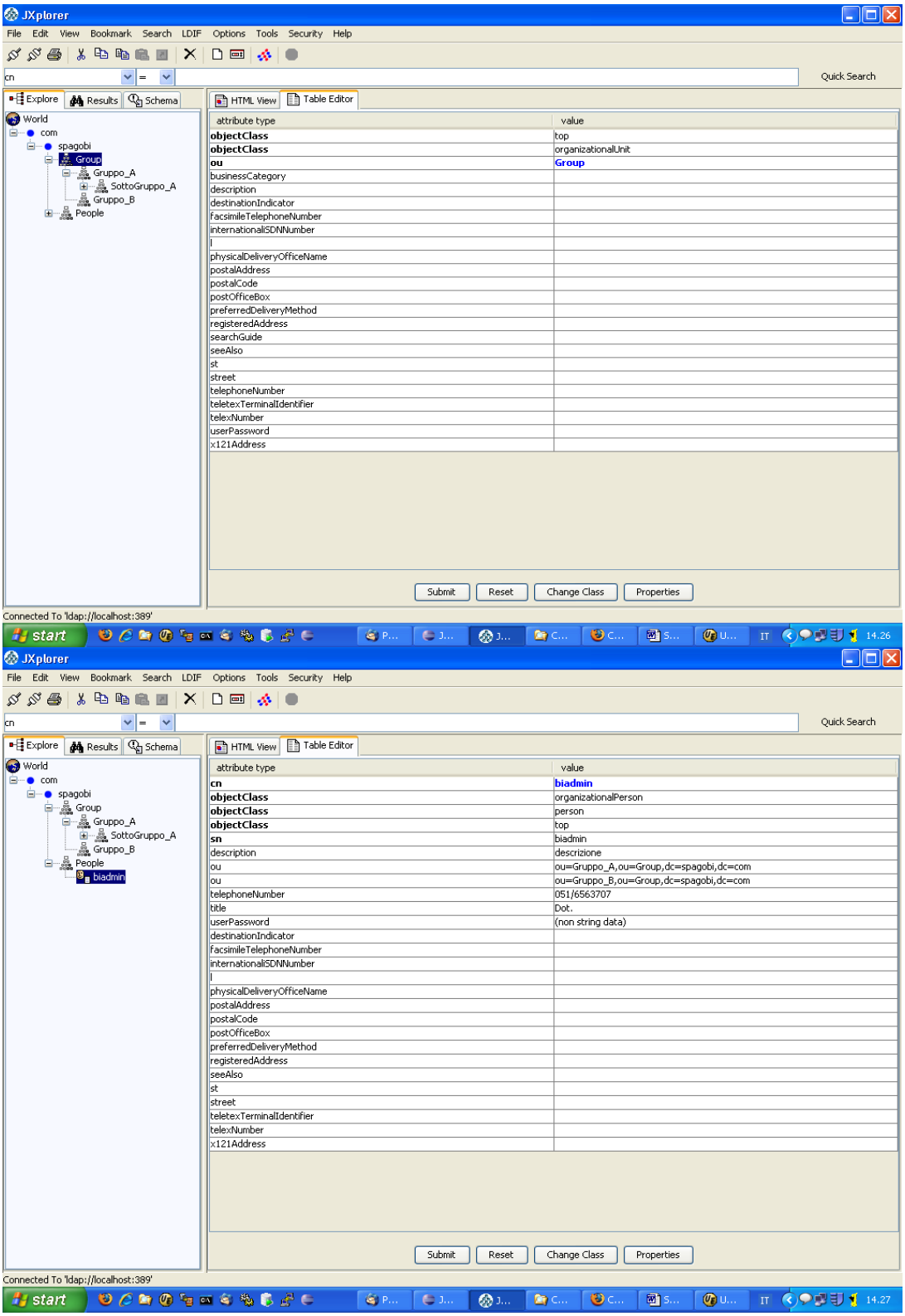
```
dn: dc=spagobi,dc=org
objectclass: dcObject
objectclass: organization
o: SpagoBI
dc: spagobi

dn: cn=root,dc= spagobi,dc= org
objectclass: organizationalRole
cn: root

# OU=People, for  users
dn:ou=People,dc=spagobi,dc=org
ou: People
objectClass: top
objectClass: organizationalUnit

# OU=Group, for groups
dn:ou=Group,dc=spagobi,dc=org
ou: Group
objectClass: top
objectClass: organizationalUnit
```

At this point we can add groups and users.
Just for example we can add the groups “Gruppo_A” and “Gruppo_B” and the user “biadmin” (see figures below).



The user “biadmin” is associated to the groups by specifing the DN of the group in the “ou” attribute: for example ou=Gruppo_A, ou=Group,dc=spagobi, dc=com

An example of .ldif file with a definition of “bidev” user is the following:

dn: uid=bidev,ou=People,dc=spagobi,dc=org
objectClass: top
objectClass: person
objectClass: organizationalPerson
ou: ou=Gruppo_A,ou=Group,dc=spagobi,dc=com
cn: bidev
sn: bidev
givenName: bidev
mail: bidev@spagobi.org
userPassword: ****

[Inside TOMCAT-INST-FILES\ldif you find an example .ldif schema.]

When you have decided the groups/users structure on LDAP system, you have to verify that configuration files are suitable for the previous choices.
Edit JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/sbiwa_ldap_authorizations.xml and check the information it contains:

```
<CONFIG>
        <USER_DN>cn=*,ou=People,dc=spagobi,dc=com</USER_DN>
        <ADMIN_USER>cn=root,dc=spagobi,dc=com</ADMIN_USER>
        <ADMIN_PSW>*****</ADMIN_PSW>
        <ATTRIBUTES_ID name="nome">description</ATTRIBUTES_ID>
        <ATTRIBUTES_ID name="cognome">sn</ATTRIBUTES_ID>
        <ATTRIBUTES_ID name="userId">cn</ATTRIBUTES_ID>
        <ATTRIBUTES_ID name="titolo">title</ATTRIBUTES_ID>
        <ATTRIBUTES_ID name="telefono">telephoneNumber</ATTRIBUTES_ID>
        <HOST>localhost</HOST>
        <PORT>389</PORT>
        <OBJECTCLASS>person</OBJECTCLASS>
        <SEARCH_ROOT>ou=People,dc=spagobi,dc=com</SEARCH_ROOT>
        <OU_ATTRIBUTE>ou</OU_ATTRIBUTE>
        <SEARCH_ROOT_GROUP>ou=Group,dc=spagobi,dc=com</SEARCH_ROOT_GROUP>
        <OBJECTCLASS_GROUP>organizationalUnit</OBJECTCLASS_GROUP>
        <ATTRIBUTES_ID_GROUP>description</ATTRIBUTES_ID_GROUP>
        <ATTRIBUTES_ID_GROUP>OU</ATTRIBUTES_ID_GROUP>
</CONFIG>
```

where:
USER_DN: it is the users' DN format, “*” character will be substitued by user identifier
ADMIN_USER: administration LDAP user, used for connection
ATTRIBUTES_ID: attributes list that will be retrieved by the LDAP system and loaded into user profile
HOST: server that hosts the LDAP system
PORT: connection port
OBJECTCLASS: class that is used for users research

SEARCH_ROOT: initial path for users research
OU_ATTRIBUTE: name of the user attribute that identifies the belonging group
SEARCH_ROOT_GROUP: initial path for groups research
OBJECT_CLASS_GROUP: class that is used for groups research
ATTRIBUTES_ID_GROUP: groups attributes

Note: for security reason, the administration password characters must be edited using Spago "DefaultCipher" class.

SpagoBI verifies if a user role is authorized to perform a certain action: in order to permit this, you have to associate each role to its set of permitted functionalities in file TOMCAT_HOME\webapps\spagobi\WEB-INF\conf\sbiwa_ldap_authorizations.xml:

```
<ENTITIES>
  <FUNCTIONALITIES>
    <FUNCTIONALITY functionalityName="EnginesManagement" description="EnginesManagement" />
    <FUNCTIONALITY functionalityName="FunctionalitiesManagement"
                   description="FunctionalitiesManagement" />
    <FUNCTIONALITY functionalityName="LovsManagement" description="LovsManagement" />
    <FUNCTIONALITY functionalityName="ConstraintManagement" description="ConstraintManagement" />
    <FUNCTIONALITY functionalityName="ParameterManagement" description="ParameterManagement" />
    <FUNCTIONALITY functionalityName="DocumentManagement" description="DocumentManagement" />
  </FUNCTIONALITIES>
</ENTITIES>
<RELATIONS>
  <PRIVILEDGES>
    <PRIVILEDGE roleName="sbi_admin" functionalityName="EnginesManagement" />
    <PRIVILEDGE roleName="sbi_admin" functionalityName="FunctionalitiesManagement" />
    <PRIVILEDGE roleName="sbi_admin" functionalityName="LovsManagement" />
    <PRIVILEDGE roleName="sbi_admin" functionalityName="ConstraintManagement" />
    <PRIVILEDGE roleName="sbi_admin" functionalityName="ParameterManagement" />
    <PRIVILEDGE roleName="sbi_admin" functionalityName="DocumentManagement" />
  </PRIVILEDGES>
</RELATIONS>
```

Edit the file spagobi.xml contained in JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/spagobi and change the security configuration (<SECURITY> tag) to:

```
<SECURITY>
  <PORTAL-SECURITY-INIT-CLASS>
    it.eng.spagobi.security.init.LdapSecurityProviderInit
  </PORTAL-SECURITY-INIT-CLASS>
  <PORTAL-SECURITY-CLASS className="it.eng.spagobi.security.LdapSecurityProviderImpl">
    <CONFIG />
  </PORTAL-SECURITY-CLASS>
  <USER-PROFILE-FACTORY-CLASS className="it.eng.spagobi.security.LdapUserProfileFactoryImpl" />
  <ROLE-NAME-PATTERN-FILTER>.*</ROLE-NAME-PATTERN-FILTER>
</SECURITY>
```

Please note that you need to put inside the spagobi/WEB-INF/lib directory the jar of the ldap security provider implementation: download the file SpagoBILdapSecurityProvider-bin-1.9.4_xxx.zip from SpagoBI donwload page and extract the contained file sbi.security.ldap-1.9.4.jar into JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/lib.

4.5.2 Usign xml files

The provided solution reads user and goups configuration from xml files and it was developed only for test purposes.
Users and groups (with their associations) are defined in file JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/sbiwa_authorizations.xml. In this file you can associate also roles with functionalities, as seen for Ldap authorization. Here you find an example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<AUTHORIZATIONS default="FALSE">
  <ENTITIES>
    <USERS>
      <USER userID="biadmin" password="biadmin" />
      <USER userID="bidev" password="bidev" />
    </USERS>
    <ROLES>
      <ROLE roleName="/spagobi/admin" description="/spagobi/admin" />
      <ROLE roleName="/spagobi/dev" description="/spagobi/dev" />
    </ROLES>
    <RESOURCES></RESOURCES>
    <APPLICATIONS></APPLICATIONS>
  </ENTITIES>
  <RELATIONS>
    <BEHAVIOURS>
      <BEHAVIOUR userID="biadmin" roleName="/spagobi/admin" />
      <BEHAVIOUR userID="bidev" roleName="/spagobi/dev" />
    </BEHAVIOURS>
    <PRIVILEDGES>
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="EnginesManagement" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="FunctionalitiesManagement" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="LovsManagement" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="ConstraintManagement" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="ParameterManagement" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="DocumentAdministration" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="ImportExport" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="Scheduler" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="Events" />
      <PRIVILEDGE roleName="/spagobi/admin" functionalityName="WorkspaceExec" />
      <PRIVILEDGE roleName="/spagobi/dev" functionalityName="LovsManagement" />
      <PRIVILEDGE roleName="/spagobi/dev" functionalityName="ConstraintManagement" />
      <PRIVILEDGE roleName="/spagobi/dev" functionalityName="ParameterManagement" />
      <PRIVILEDGE roleName="/spagobi/dev" functionalityName="DocumentAdministration" />
    </PRIVILEDGES>
  </RELATIONS>
</AUTHORIZATIONS>
```

To activate this feature to have user and goups configuration in XML files,do the followings steps :

a) Only if you haven't already done it,you need to put inside the spagobi/WEB-INF/lib directory the jar of the xml security provider implementation: download the file SpagoBIXmlSecurityProvider-bin-1.9.4_xxx.zip from SpagoBI donwload page (http://forge.objectweb.org/project/showfiles.php?group_id=204) and extract the contained file sbi.security.xml-1.9.4.jar into JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/lib.

b) Make sure that sbiwa_ldap_authorizations.xml file is disabled and that sbiwa_authorizations.xml file is enabled in the file
JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/master.xml:
...

```
<CONFIGURATOR path="/WEB-INF/conf/sbiwa_authorizations.xml" />
<CONFIGURATOR path="/WEB-INF/conf/sbiwa_menu.xml" />
<!-- <CONFIGURATOR path="/WEB-INF/conf/sbiwa_ldap_authorizations.xml" /> -->
<!-- END WEB APPLICATION -->
```

c) Edit the file spagobi.xml contained in JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/spagobi and change the security configuration (<SECURITY> tag) to:

```
<SECURITY>
  <PORTAL-SECURITY-INIT-CLASS>
    it.eng.spagobi.security.init.XmlSecurityProviderInit
  </PORTAL-SECURITY-INIT-CLASS>
  <PORTAL-SECURITY-CLASS className="it.eng.spagobi.security.XmlSecurityProviderImpl">
    <CONFIG />
  </PORTAL-SECURITY-CLASS>
  <USER-PROFILE-FACTORY-CLASS className="it.eng.spagobi.security.XmlUserProfileFactoryImpl" />
  <ROLE-NAME-PATTERN-FILTER>.*</ROLE-NAME-PATTERN-FILTER>
</SECURITY>
```

- More detailed step to change the security provider

Uncomment this part :

<!-- Xml (Spago framework) security Provider -->

<!--
-->
<SECURITY>

<PORTAL-SECURITY-INIT-CLASS>it.eng.spagobi.security.init.XmlSecurityProviderInit</PORTAL-SECURITY-INIT-CLASS>

<PORTAL-SECURITY-CLASS className="it.eng.spagobi.security.XmlSecurityProviderImpl">

<CONFIG />

</PORTAL-SECURITY-CLASS>

<USER-PROFILE-FACTORY-CLASS className="it.eng.spagobi.security.XmlUserProfileFactoryImpl" />

<ROLE-NAME-PATTERN-FILTER>.*</ROLE-NAME-PATTERN-FILTER>

</SECURITY>

<!--
-->

Comment this part :

<!-- Exo security Provider -->

<!--

```
<SECURITY>

                    <PORTAL-SECURITY-INIT-
CLASS>it.eng.spagobi.security.init.ExoPortalSecurityProviderInit</PORTAL-SECURITY-INIT-CLASS>

                    <PORTAL-SECURITY-CLASS
className="it.eng.spagobi.security.ExoGroupAsRoleSecurityProviderImpl">

                        <CONFIG>


                <NAME_PORTAL_APPLICATION>portal</NAME_PORTAL_APPLICATION>

                        </CONFIG>

                    </PORTAL-SECURITY-CLASS>

                    <USER-PROFILE-FACTORY-CLASS
className="it.eng.spagobi.security.ExoGroupAsRoleUserProfileFactoryImpl">

                    </USER-PROFILE-FACTORY-CLASS>

                <ROLE-NAME-PATTERN-FILTER>.*</ROLE-NAME-PATTERN-FILTER>

                 </SECURITY>
                -->
```

4.6 Install Engines

4.6.1 Install SpagoBIJasperReportEngine

▪Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagoBIJasperReportEngine-bin-1.9.4_xxx.zip'.Extract from the zip archive the file 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 then copy the parent SpagoBIJasperReportEngine.war directory into JBOSS-HOME/server/default/deploy folder.
▪Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIJasperReportEngine.war/WEB-INF/lib
- ○log4j-1.2.8.jar (file log4j.jar already available in Jboss server, version 1.2.8)

▪Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
- ○document type: Report
- ○engine type: External
- ○driver class: it.eng.spagobi.drivers.jasperreport.JasperReportDriver
- ○url: http://<server>:<port>/SpagoBIJasperReportEngine/JasperReportServlet.

4.6.2 Install SpagoBIJPivotEngine

▪Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagBIJPivotEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file 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 JBOSS-HOME/server/default/deploy folder.
▪Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIJPivotEngine.war/WEB-INF/lib
- ○log4j-1.2.8.jar (file log4j.jar already available in Jboss server, version 1.2.8)

▪Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
- ○document type: On-line analytical processing
- ○engine type: External
- ○driver class: it.eng.spagobi.drivers.jpivot.JPivotDriver
- ○url: http://<server>:<port>/ SpagoBIJPivotEngine/JPivotServlet.

4.6.3 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. Unzip the war file into a directory named SpagoBIQbeEngine.war. Copy the new directory into JBOSS-HOME/server/default/deploy folder.

- Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIQbeEngine.war/WEB-INF/lib (OPTIONAL : no log4j.properties):
  - log4j-1.2.8.jar (file log4j.jar already available in Jboss server, version 1.2.8)
- 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 web application define a new External Engine with (see chapter 4.7 Configure external engines):
  - 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)

4.6.4 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 rename it as SpagoBIBirtReportEngine.zip. Create a new folder named SpagoBIBirtReportEngine.war and unzip the file SpagoBIBirtReportEngine.zip inside it. Delete the SpagoBIBirtReportEngine.zip file and copy the parent SpagoBIBirtReportEngine.war directory into JBOSS-HOME/ server/default/deploy folder.
- Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIBirtReportEngine.war/WEB-INF/lib
  - log4j-1.2.8.jar
- Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
  - document type: Report
  - engine type: External
  - driver class: it.eng.spagobi.drivers.birt.BirtReportDriver
  - url: http://<server>:<port>/SpagoBIBirtReportEngine/BirtReportServlet.

4.6.5 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 rename it as SpagoBIWekaEngine.zip. Create a new folder named SpagoBIWekaEngine.war and unzip the file SpagoBIWekaEngine.zip inside it. Delete the SpagoBIWekaEngine.zip file and copy the parent SpagoBIWekaEngine.war directory into JBOSS-HOME/server/default/deploy folder.

▪Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIWekaEngine.war/WEB-INF/lib
- ○log4j-1.2.8.jar

▪Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
- ○document type: Data mining model
- ○engine type: External
- ○driver class: it.eng.spagobi.drivers.weka.WekaDriver
- ○url: http://<server>:<port>/SpagoBIWekaEngine/WekaServlet.

4.6.6 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 rename it as SpagoBIGeoEngine.zip. Create a new folder named SpagoBIGeoEngine.war and unzip the file SpagoBIGeoEngine.zip inside it. Delete the SpagoBIGeoEngine.zip file and copy the parent SpagoBIGeoEngine.war directory into JBOSS-HOME/server/default/deploy folder.

▪Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIGeoEngine.war/WEB-INF/lib
- ○xalan-2.4.0.jar
- ○xercesImpl.jar
- ○log4j-1.2.8.jar

▪Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
- ○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.

4.6.7 Install SpagoBIJPXMLAEngine

▪Connect to http://forge.objectweb.org/projects/spagobi, select the file tab and then download the file 'SpagBIJPXMLAEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJPXMLAEngine.war and rename it as SpagoBIJPXMLAEngine.zip. Create a new folder named SpagoBIJPXMLAEngine.war and unzip the file SpagoBIJPXMLAEngine.zip inside it. Delete the SpagoBIJPXMLAEngine.zip file and copy the parent SpagoBIJPXMLAEngine.war directory into JBOSS-HOME/server/default/deploy folder.

▪Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBIJPXMLAEngine.war/WEB-INF/lib
- ○log4j-1.2.8.jar

▪The driver is the same of SpagoBIJPivotEngine: control that the file sbi.driver.jpivot-1.9.4.jar is present inside folder JBOSS-HOME/server/default/deploy/exoplatform.sar/ spagobi.war/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 JBOSS-HOME/server/default/deploy/exoplatform.sar/ spagobi.war/WEB-INF/lib.

- Using the SpagoBI administration web application define a new External Engine with (see chapter 4.7 Configure external engines):
  - 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.

4.6.8Install 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.war and unzip the file SpagoBITalendEngine.zip inside it. Delete the SpagoBITalendEngine.zip file and copy the parent SpagoBITalendEngine.war directory into JBOSS-HOME/server/default/deploy folder.
- Remove the following jar files from JBOSS-HOME/server/default/deploy/ SpagoBITalendEngine.war/WEB-INF/lib
  - log4j-1.2.8.jar
- Edit file talend.properties in JBOSS-HOME/server/default/deploy/SpagoBITalendEngine.war / 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 JBOSS-HOME/server/default/deploy/ SpagoBITalendEngine.war/WEB-INF/classes and adjust your Perl installation directory (mandatory if you want to executed Perl based jobs);(*)
- Edit file talend.java.properties in JBOSS-HOME/server/default/deploy/ SpagoBITalendEngine.war/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 web application define a new External Engine with (see chapter 4.7 Configure external engines):

- document type: ETL
- engine type: External
- driver class: it.eng.spagobi.drivers.talend.TalendDriver
- 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).

4.7Configure the engines

After the execution of the previous steps you should be able to connect to web application to define the engines web application pages.

First, if you use xml file for authentication and authentication, you need to add more informations in your JBOSS-HOME/server/default/deploy/spagobi.war/WEB-INF/conf/sbiwa_authorizations.xml file. We add 4 accounts with differents rights for using SpagoBI.

Change your sbiwa_authorizations.xml file like this :

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<AUTHORIZATIONS default="FALSE">

        <ENTITIES>

            <USERS>

              <!--

                    <USER userID="admin" password="0DPiKuNIrrVmD8IUCuw1hQxNqZc=" />

                    -->

                    <USER userID="admin" password="admin" />

                    <USER userID="biadmin" password="biadmin" />

                    <USER userID="bitest" password="bitest" />

                    <USER userID="bidev" password="bidev" />

                    <USER userID="biuser" password="biuser" />

            </USERS>

            <ROLES>

                    <ROLE roleName="Administrator" description="Administrator" />

                    <ROLE roleName="/spagobi/admin" description="/spagobi/admin" />

                    <ROLE roleName="bi_admin" description="bi_admin" />
```

```
                <ROLE roleName="/spagobi/biadmin" description="/spagobi/biadmin" />

                <ROLE roleName="bi_test" description="bi_test" />

                <ROLE roleName="/spagobi/bitest" description="/spagobi/bitest" />

                <ROLE roleName="bi_dev" description="bi_dev" />

                <ROLE roleName="/spagobi/bidev" description="/spagobi/bidev" />

                <ROLE roleName="bi_user" description="bi_user" />

                <ROLE roleName="/spagobi/biuser" description="/spagobi/biuser" />

            </ROLES>

            <FUNCTIONALITIES>

                <FUNCTIONALITY functionalityName="EnginesManagement"
description="EnginesManagement" />

                <FUNCTIONALITY functionalityName="FunctionalitiesManagement"
description="FunctionalitiesManagement" />

                <FUNCTIONALITY functionalityName="LovsManagement"
description="LovsManagement" />

                <FUNCTIONALITY functionalityName="ConstraintManagement"
description="ConstraintManagement" />

                <FUNCTIONALITY functionalityName="ParameterManagement"
description="ParameterManagement" />

                <FUNCTIONALITY functionalityName="DocumentAdministration"
description="DocumentAdministration" />

                <FUNCTIONALITY functionalityName="DocumentDevelopment"
description="DocumentDevelopment" />

                <FUNCTIONALITY functionalityName="DocumentTest"
description="DocumentTest" />

                <FUNCTIONALITY functionalityName="DocumentExecution"
description="DocumentExecution" />

                <FUNCTIONALITY functionalityName="ImportExport" description="ImportExport"
/>

                <FUNCTIONALITY functionalityName="Scheduler" description="Scheduler" />
```

```
                <FUNCTIONALITY functionalityName="Events" description="Events" />

                <FUNCTIONALITY functionalityName="Worklist" description="Worklist" />

                <FUNCTIONALITY functionalityName="WorkspaceExec" description="WorkspaceExec" />

                <FUNCTIONALITY functionalityName="MapCatalogue" description="MapCatalogue" />

            </FUNCTIONALITIES>

            <RESOURCES></RESOURCES>

            <APPLICATIONS></APPLICATIONS>

        </ENTITIES>

        <RELATIONS>

            <BEHAVIOURS>

                <BEHAVIOUR userID="admin" roleName="sbi_admin" />

                <BEHAVIOUR userID="admin" roleName="/spagobi/admin" />

                <BEHAVIOUR userID="biadmin" roleName="sbi_admin" />

                <BEHAVIOUR userID="biadmin" roleName="/spagobi/biadmin" />

                <BEHAVIOUR userID="bitest" roleName="sbi_test" />

                <BEHAVIOUR userID="bitest" roleName="/spagobi/bitest" />

                <BEHAVIOUR userID="bidev" roleName="sbi_dev" />

                <BEHAVIOUR userID="bidev" roleName="/spagobi/bidev" />

                <BEHAVIOUR userID="biuser" roleName="sbi_user" />

                <BEHAVIOUR userID="biuser" roleName="/spagobi/biuser" />

            </BEHAVIOURS>

            <PRIVILEDGES>

                <PRIVILEDGE roleName="sbi_admin" functionalityName="EnginesManagement"/>
```

<PRIVILEDGE roleName="sbi_admin" functionalityName="FunctionalitiesManagement"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="LovsManagement"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="ConstraintManagement"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="ParameterManagement"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="DocumentAdministration"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="ImportExport"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="Scheduler"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="Events"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="Worklist"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="WorkspaceExec"/>

<PRIVILEDGE roleName="sbi_admin" functionalityName="MapCatalogue"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="LovsManagement"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="ConstraintManagement"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="ParameterManagement"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="DocumentDevelopment"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="Events"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="Worklist"/>

<PRIVILEDGE roleName="sbi_dev" functionalityName="WorkspaceExec"/>

<PRIVILEDGE roleName="sbi_test" functionalityName="DocumentTest"/>

<PRIVILEDGE roleName="sbi_test" functionalityName="Events"/>

<PRIVILEDGE roleName="sbi_test" functionalityName="Worklist"/>

<PRIVILEDGE roleName="sbi_test" functionalityName="WorkspaceExec"/>

<PRIVILEDGE roleName="sbi_user" functionalityName="DocumentExecution"/>

```
                <PRIVILEDGE roleName="sbi_user" functionalityName="Events"/>
                <PRIVILEDGE roleName="sbi_user" functionalityName="Worklist"/>
                <PRIVILEDGE roleName="sbi_user" functionalityName="WorkspaceExec"/>
            </PRIVILEDGES>
        </RELATIONS>
</AUTHORIZATIONS>
```

Now, you should be able to connect to the url http://localhost:8080/spagobi/index.jsp. To configure the engines

The web application defines five users:

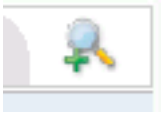
- admin (password=admin): his pages contains the SpagoBI administration web application
- biadmin (password=biadmin): his pages contains also the SpagoBI administration web application

- obidev (password=bidev): his pages contains the SpagoBI development web application
- obitest (password=bitest): his page contains the SpagoBI execution web application with the possibilty to view documents in test state
- obiuser (password=biuser): his page contains the SpagoBI execution web application

For configuring the external engines, use the biadmin for the connection.

Click on this icon (external engines):

Next click on this icon to add a new external engine :



Configure a new external engine like Jasper :