
SpagoBI Web Application Tomcat Installation Manual 1.0.0

Authors

Bernabei Angelo
Zerbetto Davide

Index

1 VERSION.....	3
2 DOCUMENT GOAL.....	3
3 REFERENCES.....	3
4 INSTALL SPAGOBİ	3
4.1 INSTALL ADDITIONAL LIBRARIES AND CONFIGURATION FILES	4
4.2 INSTALL THE METADATA DATABASE.....	4
4.2.1 Install the database driver.....	5
4.2.2 Configuring Datasource as JNDI Resource.....	5
4.3 INSTALL CMS REPOSITORY.....	6
4.4 INSTALL SPAGOBİ PLATFORM.....	6
4.4.1 Configure Hibernate.....	7
4.4.2 Configure Quartz.....	8
4.5 AUTHENTICATION AND AUTHORIZATION.....	9
4.5.1 Usign Ldap system.....	9
4.5.2 Usign xml files.....	12
4.6 INSTALL ENGINES.....	15
4.6.1 Install SpagoBIJasperReportEngine.....	15
4.6.2 Install SpagoBIJPivotEngine.....	15
4.6.3 Install SpagoBIQbeEngine.....	15
4.6.4 Install SpagoBIBirtReportEngine.....	16
4.6.5 Install SpagoBIWekaEngine.....	16
4.6.6 Install SpagoBIGeoEngine.....	16
4.6.7 Install SpagoBIJPXMLAEngine.....	17
4.6.8 Install SpagoBITalendEngine.....	17

1 Version

Version/Release n° :	1.0.0	Data Version/Release :	February, 14th 2008
Update description:	SpagoBI 1.9.4 Web Application installation steps on a Tomcat Server 6.0.x		

2 Document goal

This document provides a step by step description for SpagoBI 1.9.4 Web Application installation and configuration on a Tomcat Server 6.0.x. To get a working copy of Tomcat Server 6.0.x you can download, from Tomcat download page, the file apache-tomcat-6.0.x.zip and simply unzip it into your file system.

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>)
- 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 a Tomcat 6.0.x server; in the following we will refer to the Tomcat base directory as **TOMCAT-HOME**.

To test the Tomcat installation go to **TOMCAT-HOME/bin** and run the command **startup.bat run** (in a windows environment) or **startup.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/>, the Tomcat default 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 'spagobi-web', 'tomcat-server', 'jboss-server' and other folders. 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**.

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 **TOMCAT-INST-FILES/lib** inside your **TOMCAT-HOME/lib** directory (*).
2. Copy the **TOMCAT-INST-FILES/sbdata** folder into your **TOMCAT-HOME**. The folder contains the hsqldb metadata database and the root folder of the cms repository.

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

- ehcache-1.1.jar
- jackrabbit-core-1.0.1.jar
- slf4j-log4j12.jar
- xercesImpl.jar
- hibernate-3.1rc2.jar
- jcr-1.0.jar
- cglib-2.1_2.jar
- asm-1.5.3.jar
- portlet-api-1.0.jar
- commons-digester-1.6.jar
- jdt-compiler-3.1.1.jar
- commons-collections-3.1.jar
- dom4j-1.6.jar
- log4j-1.2.8.jar
- concurrent-1.3.4.jar
- commons-logging-api-1.0.4.jar
- jta-1.0.1B.jar
- ldap.jar

4.2 Install the Metadata Database

SpagoBI metadata are stored in a database (for this release SpagoBI supports PostgreSQL, 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 **TOMCAT-HOME/sbdata/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 **TOMCAT-HOME/sbdata/database/start.bat** (on a windows platform) or **TOMCAT-HOME/sbdata/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 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 PostgreSQL, 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 **TOMCAT-HOME/lib**. If you decide to use the SpagoBI hsqldb database you have to copy hsqldb1_8_0_2.jar from **TOMCAT-HOME/sbdata/database** to **TOMCAT-HOME/lib**.

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 **TOMCAT-INST-FILES/conf/server.xml**.
- Search for the string "<Resource name='jdbc/spagobi'";

- Copy entirely the `<Resource name="jdbc/spagobi" >` tag
- Edit the file `TOMCAT-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 `TOMCAT-INST-FILES/conf/server.xml`.
- Search for the string `"<Resource name="cms/spagobicms""`;
- Copy entirely the `<Resource name="cms/spagobicms" ... >` tag
- Edit the file `TOMCAT-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 `'${SERVER_HOME}'` with the path of your Tomcat installation directory. Example: if your Tomcat is installed into `C:\Programs\apache-tomcat` you have to replace `'${SERVER_HOME}'` with `'/Programs/apache-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 `TOMCAT-HOME/webapps` folder.

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

Edit the file spagobi.xml contained in TOMCAT-HOME/webapps/spagobi/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.

Edit the file initializer.xml contained into TOMCAT-HOME/webapps/spagobi/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 TOMCAT-HOME/webapps/spagobi/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>
-->
[....]
```

4.4.1 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 TOMCAT-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 TOMCAT-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 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)

4.5 Authentication and authorization

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

4.5.1 Usign 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
```

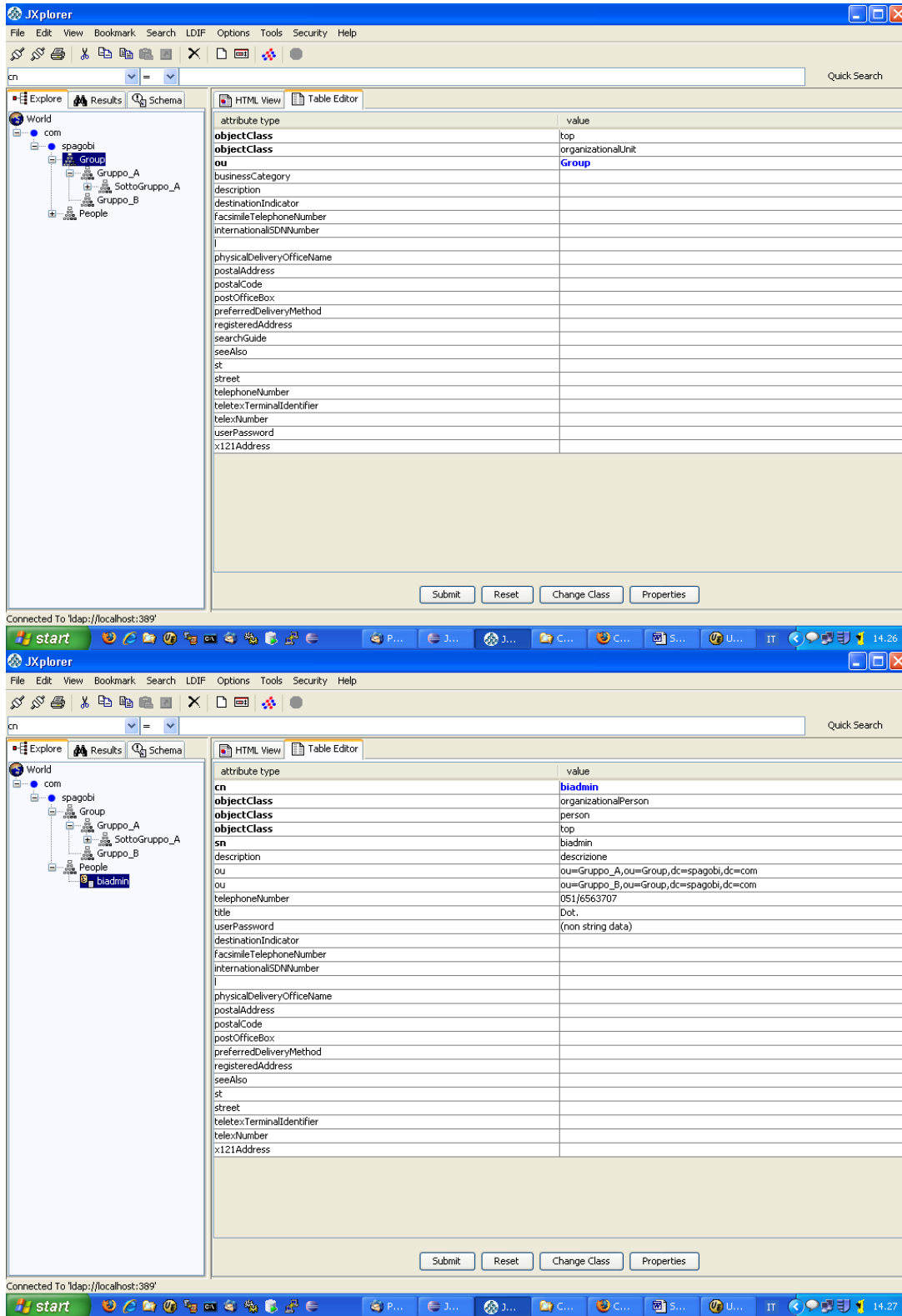
```
dn: cn=root,dc= spagobi,dc= org
objectclass: organizationalRole
cn: root
```

```
# 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).



Connected To 'ldap://localhost:389'

start

JXplorer

File Edit View Bookmark Search LDIF Options Tools Security Help

cn

Quick Search

Explore Results Schema

HTML View Table Editor

attribute type	value
objectClass	top
objectClass	organizationalUnit
ou	Group
businessCategory	
description	
destinationIndicator	
facsimileTelephoneNumber	
internationalSDNNumber	
physicalDeliveryOfficeName	
postalAddress	
postalCode	
postOfficeBox	
preferredDeliveryMethod	
registeredAddress	
searchGuide	
seeAlso	
st	
street	
telephoneNumber	
teletexTerminalIdentifier	
telexNumber	
userPassword	
x121Address	

Submit Reset Change Class Properties

Connected To 'ldap://localhost:389'

start

JXplorer

File Edit View Bookmark Search LDIF Options Tools Security Help

cn

Quick Search

Explore Results Schema

HTML View Table Editor

attribute type	value
cn	biadmin
objectClass	organizationalPerson
objectClass	person
objectClass	top
sn	biadmin
description	descrizione
ou	ou=Gruppo_A,ou=Group,dc=spagobi,dc=com
ou	ou=Gruppo_B,ou=Group,dc=spagobi,dc=com
telephoneNumber	051/6563707
title	Dot.
userPassword	(non string data)
destinationIndicator	
facsimileTelephoneNumber	
internationalSDNNumber	
physicalDeliveryOfficeName	
postalAddress	
postalCode	
postOfficeBox	
preferredDeliveryMethod	
registeredAddress	
seeAlso	
st	
street	
teletexTerminalIdentifier	
telexNumber	
x121Address	

Submit Reset Change Class Properties

Connected To 'ldap://localhost:389'

The user “biadmin” is associated to the groups by specifying 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 TOMCAT_HOME\webapps\spagobi\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 substituted 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>
    <PRIVILEGE roleName="sbi_admin" functionalityName="EnginesManagement" />
    <PRIVILEGE roleName="sbi_admin" functionalityName="FunctionalitiesManagement" />
    <PRIVILEGE roleName="sbi_admin" functionalityName="LovsManagement" />
    <PRIVILEGE roleName="sbi_admin" functionalityName="ConstraintManagement" />
    <PRIVILEGE roleName="sbi_admin" functionalityName="ParameterManagement" />
    <PRIVILEGE roleName="sbi_admin" functionalityName="DocumentManagement" />
  </PRIVILEDGES>
</RELATIONS>
```

Edit the file spagobi.xml contained in TOMCAT-HOME/webapps/spagobi/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 download page and extract the contained file sbi.security.ldap-1.9.4.jar into TOMCAT-HOME/webapps/spagobi/WEB-INF/lib.

4.5.2 Usign xml files

The provided solution reads user and groups configuration from xml files and it was developed only for test purposes.

Users and groups (with their associations) are defined in file TOMCAT-HOME/webapps/spagobi/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>
    <PRIVILEGES>
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="EnginesManagement" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="FunctionalitiesManagement" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="LowsManagement" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="ConstraintManagement" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="ParameterManagement" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="DocumentAdministration" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="ImportExport" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="Scheduler" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="Events" />
      <PRIVILEGE roleName="/spagobi/admin" functionalityName="WorkspaceExec" />
      <PRIVILEGE roleName="/spagobi/dev" functionalityName="LowsManagement" />
      <PRIVILEGE roleName="/spagobi/dev" functionalityName="ConstraintManagement" />
      <PRIVILEGE roleName="/spagobi/dev" functionalityName="ParameterManagement" />
      <PRIVILEGE roleName="/spagobi/dev" functionalityName="DocumentAdministration" />
    </PRIVILEGES>
  </RELATIONS>
</AUTHORIZATIONS>
```

Make sure that sbiwa_authorizations.xml file is referenced from TOMCAT-HOME/webapps/spagobi/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 -->
```

Edit the file spagobi.xml contained in TOMCAT-HOME/webapps/spagobi/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>
```

Please note that 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 download page and extract the contained file sbi.security.xml-1.9.4.jar into TOMCAT-HOME/webapps/spagobi/WEB-INF/lib.

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 'SpagoBIJasperReport-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJasperReportEngine.war and copy it into the TOMCAT-HOME/webapps folder;
- copy the file SpagoBIJasperReportEngine.xml from TOMCAT-INST-FILES/conf/Catalina/localhost into TOMCAT-HOME/conf/Catalina/localhost.
- Remove commons-beanutils-1.6.jar library from TOMCAT-HOME/webapps/SpagoBIJasperReportEngine/WEB-INF/lib 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`.

4.6.2 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 TOMCAT-HOME/webapps folder.
- Copy the file SpagoBIJPivotEngine.xml from TOMCAT-INST-FILES/conf/Catalina/localhost into TOMCAT-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`.

4.6.3 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 and copy it into the TOMCAT-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 commons-logging-1.0.4.jar library from TOMCAT-HOME/webapps/SpagoBIQbeEngine/WEB-INF/lib folder
- 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)

4.6.4 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 copy it into the TOMCAT-HOME/webapps folder.
- Copy the file SpagoBIBirtReportEngine.xml from TOMCAT-INST-FILES/conf/Catalina/localhost into TOMCAT-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`.

4.6.5 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 copy it into the TOMCAT-HOME/webapps folder.
- Copy the file SpagoBIWekaEngine.xml from TOMCAT-INST-FILES/conf/Catalina/localhost into TOMCAT-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`.

4.6.6 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 copy it into the TOMCAT-HOME/webapps folder.
- Copy the file SpagoBIGeoEngine.xml from TOMCAT-INST-FILES/conf/Catalina/localhost into TOMCAT-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`

4.6.7 Install SpagoBIJXMLAEngine

- Connect to <http://forge.objectweb.org/projects/spagobi>, select the file tab and then download the file 'SpagoBIJXMLAEngine-bin-1.9.4_xxx.zip'. Extract from the zip archive the file SpagoBIJXMLAEngine.war and copy it into the TOMCAT-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 TOMCAT-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 TOMCAT-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>/SpagoBIJXMLAEngine/JPivotServlet`

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

4.6.8 Install 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 TOMCAT-HOME/webapps folder.
- Edit file talend.properties in TOMCAT-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 TOMCAT-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 `TOMCAT-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).