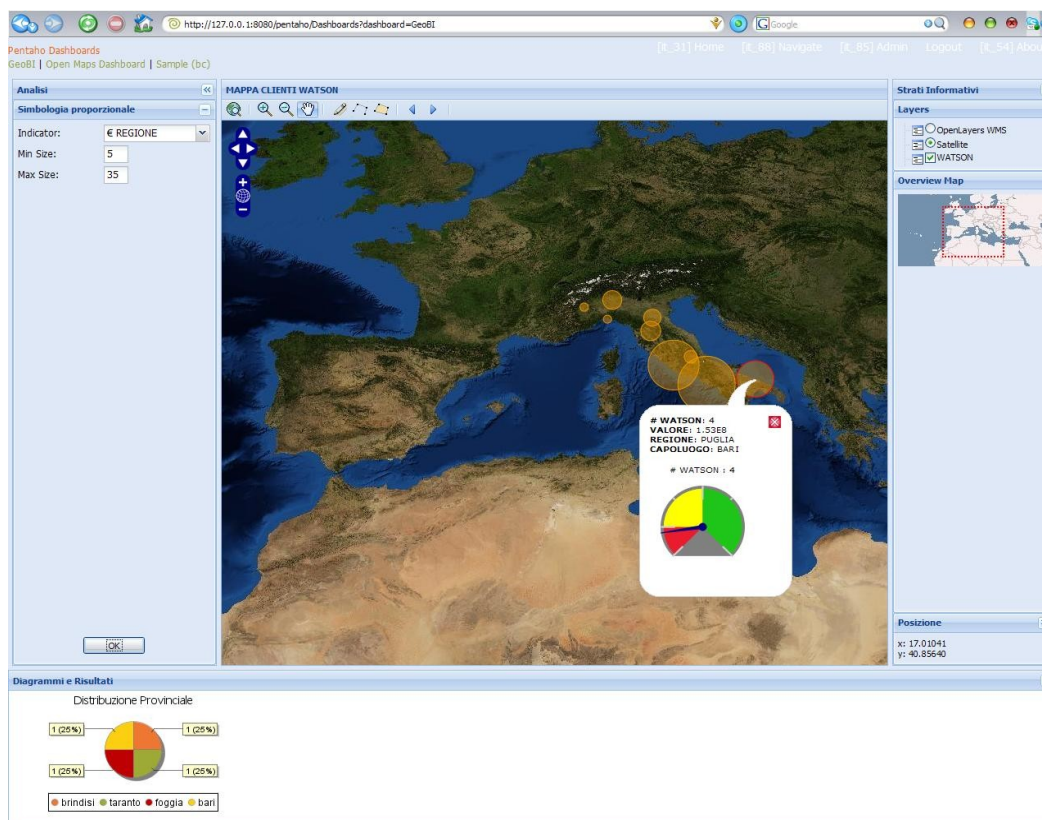


# GEOREPORT INSTALLATION NOTES



## Index

<a href="#">What need</a> .....	2
<a href="#">Download</a> .....	2
<a href="#">Install</a> .....	2
<a href="#">Notes</a> .....	5





## What need

GeoReport Module **prototype** is based on PentahoBI, GeoServer, Mapfish GeoJson lib, Mapfish Client and OpenLayers.

GeoReport is still a prototype but it gives you a method to use Business Intelligence tools and GIS technologies together in the same application!

## Download

You must download the following applications to run Georeport module.

1. PentahoBI 1.7.1 from [Sourceforge.net](http://sourceforge.net)
2. GeoServer 1.7.1 (<http://geoserver.org/display/GEOS/GeoServer+1.7.1>)
3. Mapfish 1.1 (<http://www.mapfish.org/downloads/>)  
here you can download both Client, Server and libraries (e.g. Mf-geo-lib)

NOTE: For our demo we choose Pentaho PCI (Pre Configured Install) running on JBOSS. For use it with Tomcat you need only few changes.

## Install

The next step is to install the applications above as following:

1. Install Pentaho PCI (You can follow Installation Notes [here](#)). We will refer pentaho directory installation as `${PENTAHO_DIR}` and jboss installation dir as `${JBOSS_HOME}`. At the end you can reach it such as:  
<http://localhost:8080/pentaho>
2. Install GeoServer as you want e.g.:  
<http://localhost:8080/geoserver>
3. Extract MapFish-1.1.tar.gz in a directory such as `/user/local/mapfish/`
4. Create “mapfish” directory under  
`${JBOSS_HOME}/server/default/deploy/pentaho.war/js/`
5. Copy `/Mapfish-1.1/client` under  
`${JBOSS_HOME}/server/default/deploy/pentaho.war/js/mapFish/`
6. Copy `dashboards` folder under `${PENTAHO_DIR}/pentaho-solutions/`
7. Configure new jndi name for postgres under JBOSS called **Pgsq12**, following [Pentaho instructions](#). (You can download postgres database driver [here](#)). You have to connect to

*geobi-samples* database. Your jboss datasource should be similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>Pgsql2</jndi-name>
    <connection-url>jdbc:postgresql://localhost:5432/geobi-
samples</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>username</user-name>
    <password>password</password>
  </local-tx-datasource>
</datasources>
```

9. Register Dashboard.jsp and MapOl.jsp in `${JBOSS_HOME}/server/default/deploy/pentaho.war/WEB-INF/web.xml` as following:

```
<servlet>
  <servlet-name>Dashboards</servlet-name>
  <jsp-file>/jsp/Dashboards.jsp</jsp-file>
</servlet>
<servlet>
  <servlet-name>MapOl</servlet-name>
  <jsp-file>/jsp/MapOl.jsp</jsp-file>
</servlet>
```

Then register servlet Mappings as following:

```
<servlet-mapping>
  <servlet-name>MapOl</servlet-name>
  <url-pattern>/MapOl</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Dashboards</servlet-name>
  <url-pattern>/Dashboards</url-pattern>
</servlet-mapping>
```

11. Copy Dashboard.jsp and MapOl.jsp under `${JBOSS_HOME}/server/default/deploy/pentaho.war/jsp`
12. Configure GeoServer in order to serve PostGIS data. You can insert sample data by running `geobi-samples_pgsql.sql` script
13. Create “*geobi*” directory under `${PENTAHO_DIR}/pentaho-solutions/system`
14. Copy `geobi.xml` under `${PENTAHO_DIR}/pentaho-solutions/system/geobi` or you can create `geobi.xml` file under `/pentaho-solutions/system/geobi/` as following:

```

<geobi>
  <map>NAME_OF_THE_MAP</map>
  <layer>layer-LAYER;type</layer>
  <param>indic_value1-INDIC_NAME1;indic_value2-INDIC_NAME2;ecc..</param>
  <att>ATT_NAME1-att_name1;ecc...</att>
  <geocode>id</geocode>
  <pos>integer</pos>
  <action>path;name.xaction</action>
  <url>http://server_name</url>
</geobi>

```

<b>Bolded name</b>	<b>Description</b>
<b>NAME_OF_THE_MAP</b>	map name
<b>layer</b>	is the name of layer stored in GeoServer
<b>LAYER</b>	is the name you want to display on the map Layer tree
<b>type</b>	it can be “chorop” or “prop” if you want respectively choropleths or proportional symbols
<b>indic_value</b>	is the name of the attribute used as an indicator for Geostatistical analysis
<b>INDIC_NAME</b>	is the name of the indicators that appears on the GeoStat tab at the left of the map
<b>ATT_NAME</b>	attribute name that appears on feature click
<b>att-name</b>	is the real name of the attribute of the alphanumeric resultset coming from xactions execution
<b>id</b>	is the name of the id attribute of the geometries data (e.g. PostGIS table) that link to alphanumeric resultset attribute id coming from xactions execution
<b>pos tag</b> <pos>	contains the position (integer value) of the id attribute value (linked to the id above for geometries) contained in the xaction resultset
<b>path</b>	is the relative path to the “name.xaction”
<b>name.xaction</b>	is the name of the xaction used for retrieve alphanumeric data from databases
<b>url tag</b> <url>	contains the http string to connect to GeoServer (everywhere)

15. Compile mapfish-geo-lib using mvn following the instruction under `/user/local/mapfish/mapfish/server/java/mapfish-geo-lib/`
16. Copy the compiled jar file (`mapfish-geo-lib-1.1.jar`) under `${JBOSS_HOME}/server/default/deploy/pentaho.war/WEB-INF/lib/`
17. Download and copy under `${JBOSS_HOME}/server/default/deploy/pentaho.war/WEB-INF/lib/` the following jar files:
  - ✓ json.jar (version 20080701)
  - ✓ jts.jar (version 1.8)
  - ✓ jtsio.jar (version 1.8)
18. Copy `georeport.png` under `${JBOSS_HOME}/server/default/deploy/pentaho.war/js/`
19. Run PentahoBI
20. Reload pentaho solutions repository and system settings by choosing *Admin* from <http://localhost:8080/pentaho/>
19. Click on *INOVA GeoBI Dashboard* by choosing *Navigate->Solutions->Samples->GeoReport* from <http://localhost:8080/pentaho/> or type in your browser as following:  
<http://localhost:8080/pentaho/Dashboards?dashboard=GeoBI>

## Notes

- ✓ Sample data must be installed on PostgreSQL\PostGIS, so you have to install [PostGIS](#) on PostgreSQL
- ✓ We supposed you can use GeoServer. If not you must read [here](#)
- ✓ If you do not use **Pentaho 1.7** and would use **Pentaho 1.6** consider that you must change **MapOL.jsp** module because API reference path was modified from a version to another. In particular:  
import `org.pentaho.core.connection.*` instead of `org.pentaho.commons.connection.*`
- ✓ You can modify *watson example* in order to run any others you want. To do that you must configure new [xactions](#) and modify `Dashboard.jsp` (*onFeatureSelect* and *showDetails* JavaScript functions).