

Pentaho and Liferay on GlassFish Instructions

These instructions document how to install Pentaho and Liferay on GlassFish manually. This document is divided into the following sections:

- Software Downloads
- Liferay Installation and Configuration
- Pentaho Installation and Configuration
- Pentaho / Liferay Integration
- Appendix

Software Downloads

The following software needs to be downloaded:

- GlassFish v2 UR2 (<https://glassfish.dev.java.net/downloads/v2ur2-b04.html>)
- MySQL 5.0 (<http://dev.mysql.com/downloads/mysql/5.0.html#downloads>)
 - Windows: download Windows ZIP/Setup.EXE (x86)
<http://dev.mysql.com/downloads/mysql/5.0.html#win32>
 - Solaris: download the version that matches your platform / operating system
<http://dev.mysql.com/downloads/mysql/5.0.html#solaris>
- MySQL Connector/J 5.1 (<http://dev.mysql.com/downloads/connector/j/5.1.html>)
- Liferay V4.x or 5.x (http://sourceforge.net/project/showfiles.php?group_id=49260)

The files that need to be downloaded for Liferay 4.4.2 are:

- Liferay Plugins SDK 4.4.2 (http://downloads.sourceforge.net/lportal/liferay-plugins-sdk-4.4.2.zip?modtime=1203963537&big_mirror=1)
- Liferay Portal 4.4.2 WAR (http://downloads.sourceforge.net/lportal/liferay-portal-4.4.2.war?modtime=1203963865&big_mirror=1)
- Liferay Portal Dependencies 4.4.2 (http://downloads.sourceforge.net/lportal/liferay-portal-dependencies-4.4.2.zip?modtime=1203963916&big_mirror=1)
- Liferay Portal SQL 4.4.2 (http://downloads.sourceforge.net/lportal/liferay-portal-sql-4.4.2.zip?modtime=1203965072&big_mirror=1)

The files that need to be downloaded for Liferay 5.0.1 RC are:

- Liferay Plugins SDKS 5.0.1 (http://downloads.sourceforge.net/lportal/liferay-plugins-sdk-5.0.1.zip?modtime=1208195480&big_mirror=1)
- Liferay Portal 5.0.1 WAR (http://downloads.sourceforge.net/lportal/liferay-portal-5.0.1.war?modtime=1208195587&big_mirror=1)
- Liferay Portal Dependencies 5.0.1 (http://downloads.sourceforge.net/lportal/liferay-portal-dependencies-5.0.1.zip?modtime=1208195614&big_mirror=1)

- Liferay Portal SQL 5.0.1 (http://downloads.sourceforge.net/lportal/liferay-portal-sql-5.0.1.zip?modtime=1208197000&big_mirror=1)

These files should be downloaded and unzipped into a Liferay directory on the local system. This location will be referred to as \$LIFERAY_HOME.

- Pentaho Business Intelligence Suite - 1.6.0 Open Source GA

The files that need to be downloaded for Pentaho are:

- J2EE Deployments (http://downloads.sourceforge.net/pentaho/pentaho_j2ee_deployments-1.6.0.GA.863-a.zip)
- Sample Data(http://downloads.sourceforge.net/pentaho/pentaho_sample_data-1.6.0.GA.863.zip)
- Sample Solutions (http://downloads.sourceforge.net/pentaho/pentaho_solutions-1.6.0.GA.863.zip)
- Sample Database (http://source.pentaho.org/svnroot/pentaho-data/trunk/mysql5/SampleDataDump_MySql.sql)

These files should be download and unzipped into \$PENTAHO_INSTALL_DIR. There should be three folders in this directory: data, solutions, and J2EE deployments.

- Other JAR files
 - log4j.jar version 1.2 is needed if you are using Liferay 4.x (<https://lportal.svn.sourceforge.net/svnroot/lportal/portal/trunk/lib/portal/log4j.jar>)
 - dom4j-1.6.1.jar (http://sourceforge.net/project/showfiles.php?group_id=16035&package_id=14121&release_id=328664)
 - jaxen-1.1.1.jar (<http://jaxen.codehaus.org/releases.html>)

Liferay Installation and Configuration

The following section details the software installation and configuration that needs to occur for Liferay to run in a GlassFish / MySQL environment. At a high level, the steps that need to be taken are:

1. Install and configure GlassFish
2. Install and configure MySQL
3. Configure GlassFish for Liferay
4. Configure Liferay

Each of these steps are outlined in detail below.

a) *Install and configure GlassFish*

Instructions to install and configure GlassFish can be found at:

<https://glassfish.dev.java.net/downloads/v2ur2-b04.html>

Make sure to follow the directions for the appropriate platform / operating system

b) ***Install and configure MySQL***

Instructions to install and configure MySQL can be found at:

<http://dev.mysql.com/doc/refman/5.0/en/installing-cs.html>

Make sure to follow the directions for the appropriate platform / operating system.

The next step is to configure MySQL for Liferay. To configure the minimal Liferay database environment, follow these steps:

a) To create the Liferay “lportal” database, issue the following command:

```
mysql -uroot -ppassword < $LIFERAY_HOME/liferay-portal-sql-*/create-minimal/create-minimal-mysql.sql
```

Note that the * above should be substituted with the Liferay version that was downloaded.

b) To create a specific Liferay user called “lportal” with the appropriate privileges, issue the following commands:

```
mysql -uroot -ppassword
mysql> grant all on lportal.* to lportal identified by 'lportal';
mysql> grant all on lportal.* to lportal@localhost identified by 'lportal';

mysql> flush privileges;

mysql> exit;
```

c) ***Configure GlassFish for Liferay***

Version 4.x

To configure GlassFish for Liferay version 4.x, follow the instructions located at:

<http://docs.liferay.com/4.3/official/liferay-4-administration-guide.pdf>

The manual instructions begin on page 39 of the Administration Guide. Complete the 27 steps in this document (through page 41).

In addition to the instructions listed in this guide, you will also need to copy log4j.jar, which was downloaded earlier, to \$GLASSFISH_HOME/domains/domain1/lib. GlassFish will need

to be restarted once this has occurred. At that point, you should be able to open <http://localhost:8080> in your browser and see the default Liferay home page. You can login with user id test@liferay.com and password test.

Version 5.x

To configure GlassFish for Liferay version 5.x, follow the instructions located at:

http://blogs.sun.com/sfehrman/entry/liferay_5_0_on_existing

d) ***Configure Liferay***

The following steps need to be completed to configure Liferay:

- a) Sign in to Liferay at <http://localhost:8080> as the Administrator (test@liferay.com/test).
- b) Add the “Plugin Installer” portlet if not already present. Under “Welcome, Test Test!” in the top right hand corner of the browser, click the down arrow and choose Add Application. In the Add Application search window, enter 'plugin installer'. Then, click on Add Plugin Installer.
- c) Click on the Configuration tab of the Plugin Installer and change the “Deploy Directory” to the deployment directory of your choice. Then, press “Save”.

Pentaho Business Intelligence Suite Installation and Configuration

The following section details the software installation and configuration that needs to occur for Pentaho Business Intelligence Suite to run in a GlassFish / MySQL environment. At a high level, the steps that need to be taken are:

- 1) Configure MySQL for Pentaho Business Intelligence Suite
- 2) Build Pentaho Business Intelligence Suite for GlassFish
- 3) Deploy Pentaho Business Intelligence Suite in GlassFish
- 4) Getting Pentaho and Liferay Running in GlassFish Together

Each of these steps are outlined in detail below.

1. Configure MySQL for Pentaho Business Intelligence Suite

Since MySQL is already installed for Liferay, you can use that existing installation for Pentaho Business Intelligence Suite. The steps that need to be taken are outlined below:

- a) Import the sample data into MySQL:

```
mysql -uroot -ppassword
mysql> source $PENTAHO_INSTALL_DIR\SampleDataDump_MySql.sql;
```

- b) Set up privileges for the Pentaho Business Intelligence Suite Users:

```
mysql -uroot -ppassword
mysql> grant all privileges on *.* to hibuser;
mysql> grant all privileges on *.* to pentaho_user;
mysql> grant all privileges on *.* to pentaho_admin;
mysql> set password for 'hibuser' = PASSWORD('password');
mysql> set password for 'pentaho_user' = PASSWORD('password');
mysql> set password for 'pentaho_admin' = PASSWORD('password');
mysql> flush privileges;
```

- c) Verify that you can login to MySQL as the necessary Pentaho users (hibuser, pentaho_user, and pentaho_admin) . At the command line, type:

```
mysql -u<user> -p<password>
```

If you have problems with the users, try to create the users with:

```
create user 'pentaho_user'@'localhost';
create user 'pentaho_admin'@localhost;
create user 'hibuser'@localhost;
```

2. ***Build Pentaho Business Intelligence Suite for GlassFish***

At this point, all of the necessary Pentaho files should be downloaded to \$PENTAHO_INSTALL_DIR. First, configuration changes need to be made for GlassFish and then Pentaho can be built. All of the steps are outlined below.

Note: To build Pentaho, ant must be installed on the system. ant can be downloaded from <http://ant.apache.org/bindownload.cgi>. It is also available in the GlassFish lib directory.

- a) Make the following modifications to \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/build.properties:

Add the following parameters:

- rdbms=mysql5
- ear.dir=\${build.res.dir}/ear
- tomcat5.dir=\${build.wars.dir}/tomcat/\${rdbms}
- staging.resources=pentaho-res

Change the following parameters:

- data.dir should point to \$PENTAHO_INSTALL_DIR/pentaho-data

- solutions.dir should point \$PENTAHO_INSTALL_DIR/pentaho-solutions
- b) Under \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a, create the directory structure build/pentaho-ears/orion. The ant build will place the ears in this directory.
 - c) Modify \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-webapp/WEB-INF/web.xml. Change the value of "solution-path" to the \$PENTAHO_INSTALL_DIR/pentaho-solutions. As an example, "solution-path" could be set to C:\pentaho\pentaho-solutions.
 - d) In \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-res/ear/application.xml, delete all the additional Java modules referring *-ds.xml files. The modified application.xml file is [here](#). These modules create datasources in JBoss and hence are needed only for a JBoss compatible ear. For GlassFish though, we create connection pools and data resources manually, through the admin console or by using the asadmin command-line tool.
 - e) In \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-res/orion/data-sources.xml, modify all the data source's connection-driver and url values to indicate mysql, rather than hsql. Below is an example of the Hibernate data source. Replicate it for all the other data sources in the file.

```

<data-source
class="com.evermind.sql.DriverManagerDataSource"
name="Hibernate"
location="jdbc/HibernateDS"
xa-location="jdbc/xa/HibernateDS"
ejb-location="jdbc/Hibernate"
connection-driver="com.mysql.jdbc.Driver"
username="hibuser"
password="password"
url="jdbc:mysql://localhost:3306/hibernate"
inactivity-timeout="30"
/>

```

- f) Verify that connection.username and connection.password params in \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-res/hibernate/mysql5/hibernate.cfg.xml reflect the username and password for the database at jdbc:mysql://:3306/hibernate. By default, the values are "hibuser" and "password".
- g) Copy the 3 jar files (dom4j, jaxen, mysql-connector-java), which were downloaded earlier, to \$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-third-party.

- h) Delete `pentaho-vfs.jar` from `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-third-party`.
- i) Create a directory `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/build/uncompressed`.
- j) Uncompress `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-third-party/commons-vfs-20061214.jar` into the new directory (`$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/build/uncompressed`). You can use the `jar` command from the JDK:

```
jar xvf $PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-third-party/commons-vfs-20061214.jar
```

- k) Edit the `org/apache/commons/vfs/impl/providers.xml` file in this uncompressed jar. Add the following element within the root `<providers>` element:

```
<provider class="org.pentaho.repository.filebased.solution.SolutionRepositoryVfs">  
  <scheme name="solution"/>  
</provider>
```

- l) After the modification, save the file and repackage the commons jar:

```
jar cvfm $PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-third-party/commons-vfs-20061214.jar meta-inf/manifest.mf .
```

- m) Rename the file `metadata_mysql5.xmi` to `metadata.xmi` in `$PENTAHO_INSTALL_DIR/pentaho-solutions/samples`.
- n) From `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a`, run `ant war-pentaho-tomcat-mysql`. Make sure that you see “Build Successful” before continuing.
- o) From `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a`, run `ant build-orion2.0.5-ear`. Make sure that you see “Build Successful” before continuing. After successful completion of this command, `pentaho.ear` will be created in `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/build/pentaho-ears/orion`.

3. *Deploy Pentaho Business Intelligence Suite in GlassFish*

- a) JDBC Connection Pools and Resources associated with the Pentaho Business Intelligence Suite database need to be created. This can be done either through the GlassFish Admin Console or via `asadmin` at the command line. For this scenario, pools and resources are created for Pentaho's Sample Database only (i.e. quartz, hibernate, sampledataAdmin, and sampledata). To create the pools and resources, run the following command:

```
$GLASSFISH_HOME/bin/asadmin add-resources GlassFish-pentaho-ds.xml.
```

Upon completion of this step, you should see that the add resources completed successfully. For naming the resources `jdbc/Quartz`, `jdbc/Hibernate`, `jdbc/SampleDataAdmin` and `jdbc/SampleData`, the `web.xml` syntax from above was used. `GlassFish-pentaho-ds.xml` can be found at the end of this document.

- b) Verify the connection pools using the command `asadmin ping-connection-pool <connection-pool-name>`. Connection pool names can be retrieved from the `GlassFish-pentaho-ds.xml` file.

For the ping to be successful, the MySQL Connector JAR has to be dropped into the `$GLASSFISH_HOME/lib` directory and the GlassFish server restarted. However, if you choose to skip this verification step, the connector jar in the `penatho-third-party` directory should be sufficient for the Pentaho application to establish connections to the MySQL database.

- c) Deploy the `pentaho.ear` file to GlassFish.

Command line deployment:

Deploy via `asadmin` using the command `asadmin deploy --user=<adminuser> <path to pentaho.ear>`

GUI deployment:

- Login to GlassFish Admin Server
- Browse to Common Tasks ->Deploy Enterprise Applications/Modules
- Add the following information:
 - Type = Enterprise Application (.ear)
 - Location = path to pentaho.ear file (choose to either upload or make accessible to server locally)
 - Application Name = Pentaho
 - Virtual Server = server

- Select OK
- d) To get Pentaho to work with Liferay, other Pentaho applications need to be deployed into GlassFish. Copy `pentaho-portal-layout.war`, `pentaho-style.war` and `sw-style.war` into the autodeploy directory of the GlassFish domain onto which you wish to deploy (i.e. `$GLASSFISH_HOME/domains/domain1/autodeploy`). The WAR files can be found in the `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/build\pentaho-wars` directory.
- e) To verify the deployment, go to `http://localhost:8080/pentaho/Home` and try out Pentaho by selecting one of the valid users in the drop in box and logging in.

Notes:

The above deployment only allows for access to the Pentaho application from a browser that is local to the application server instance.

To enable remote client (browser) access to the application, locate the following parameter in `$PENTAHO_INSTALL_DIR/pentaho_j2ee_deployments-1.6.0.GA.863-a/pentaho-webapp/WEB-INF/web.xml`

```
<param-name>base-url</param-name>
<param-value>http://localhost:8080/pentaho/</param-value>
```

and modify the `base-url` parameter value to the actual server name

```
<param-name>base-url</param-name>
<param-value>http://<servername>:8080/pentaho/</param-value>
```

Additionally, the Pentaho folders need to be accessible to the application server for sample data to be available to the application.

Pentaho Business Intelligence Suite and Liferay Integration

The following section details the configuration that needs to occur for Pentaho Business Intelligence Suite to work within Liferay.

1. Sign in to Liferay at <http://localhost:8080> as the Administrator (`test@liferay.com/test`).
2. Go to the Liferay portlets welcome Menu option and click “Add Application”. You will see an Application docking window containing a collection of portlets.

3. Search for Plugin Installer and select it. Import the pentaho.war file into Liferay using the Upload file option. pentaho.war can be found at \$PENTAHO_INSTALL_DIR\pentaho_j2ee_deployments-1.6.0.GA.863-a\build\pentaho-wars\tomcat\mysql5. Once pentaho.war is selected, click Install.
4. Once updated, you can search for Pentaho in Liferay's Add Application. The portlets can be dragged onto the screen.

Appendix

GlassFish-pentaho-ds.xml

```

    <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC
    "-//Sun Microsystems Inc.//DTD Application Server 9.1 Domain//EN"
    "*http://www.sun.com/software/appserver/dtds/sun-
resources_1_2.dtd*">

<resources>
    <jdbc-connection-pool name="pentahoHibernate" datasource-
classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" res-
type="javax.sql.DataSource">
        <property name="user" value="hibuser"/>
        <property name="password" value="password"/>
        <property name="url" value="jdbc:mysql://:3306/hibernate"/>
    </jdbc-connection-pool>
    <jdbc-resource enabled="true" jndi-name="jdbc/Hibernate" object-
type="user" pool-name="pentahoHibernate"/>

    <jdbc-connection-pool name="pentahoSampledata" datasource-
classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" res-
type="javax.sql.DataSource">
        <property name="user" value="pentaho_user"/>
        <property name="password" value="password"/>
        <property name="url"
value="jdbc:mysql://:3306/sampledata"/>
    </jdbc-connection-pool>
    <jdbc-resource enabled="true" jndi-name="jdbc/SampleData" object-
type="user" pool-name="pentahoSampledata"/>

```

```
<jdbc-connection-pool name="pentahoSampledataAdmin" datasource-
classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" res-
type="javax.sql.DataSource">
  <property name="user" value="pentaho_admin"/>
  <property name="password" value="password"/>
  <property name="url"
value="jdbc:mysql://:3306/sampledata"/>
</jdbc-connection-pool>
<jdbc-resource enabled="true" jndi-name="jdbc/SampleDataAdmin"
object-type="user" pool-name="pentahoSampledataAdmin"/>

  <jdbc-connection-pool name="pentahoQuartz" datasource-
classname="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" res-
type="javax.sql.DataSource">
  <property name="user" value="pentaho_user"/>
  <property name="password" value="password"/>
  <property name="url" value="jdbc:mysql://:3306/quartz"/>
</jdbc-connection-pool>
<jdbc-resource enabled="true" jndi-name="jdbc/Quartz" object-
type="user" pool-name="pentahoQuartz"/>

</resources>
```