

PostgresqlPlugin Framework

To use PostgreSQL as the persistence layer with WebObjects, you need the PostgreSQL JDBC driver and the PostgresqlPlugin framework from Project Wonder.

JDBC Driver

A JDBC driver is required for WebObjects, just like for any Java application that talks to a database. You should download the driver that matches the version of PostgreSQL that you have installed from PostgreSQL's [JDBC driver download page](#).

There are several ways to install the driver:

1. Put the JAR in the JRE's extensions directory: `/Library/Java/Extensions` on Mac OS X, or `$JAVA_HOME/lib/ext` on Unix. (Be aware that putting JARs in the extensions directory carries caveats with respect to classpath references and versioning. It is loaded by the system classloader and is a global version for all of your Java applications.)
2. Put the JAR in the `Libraries` directory of a specific application.
3. Create a framework containing the JAR in `Libraries`, and then reference that framework from your applications.

JDBC driver adjusts timestamps

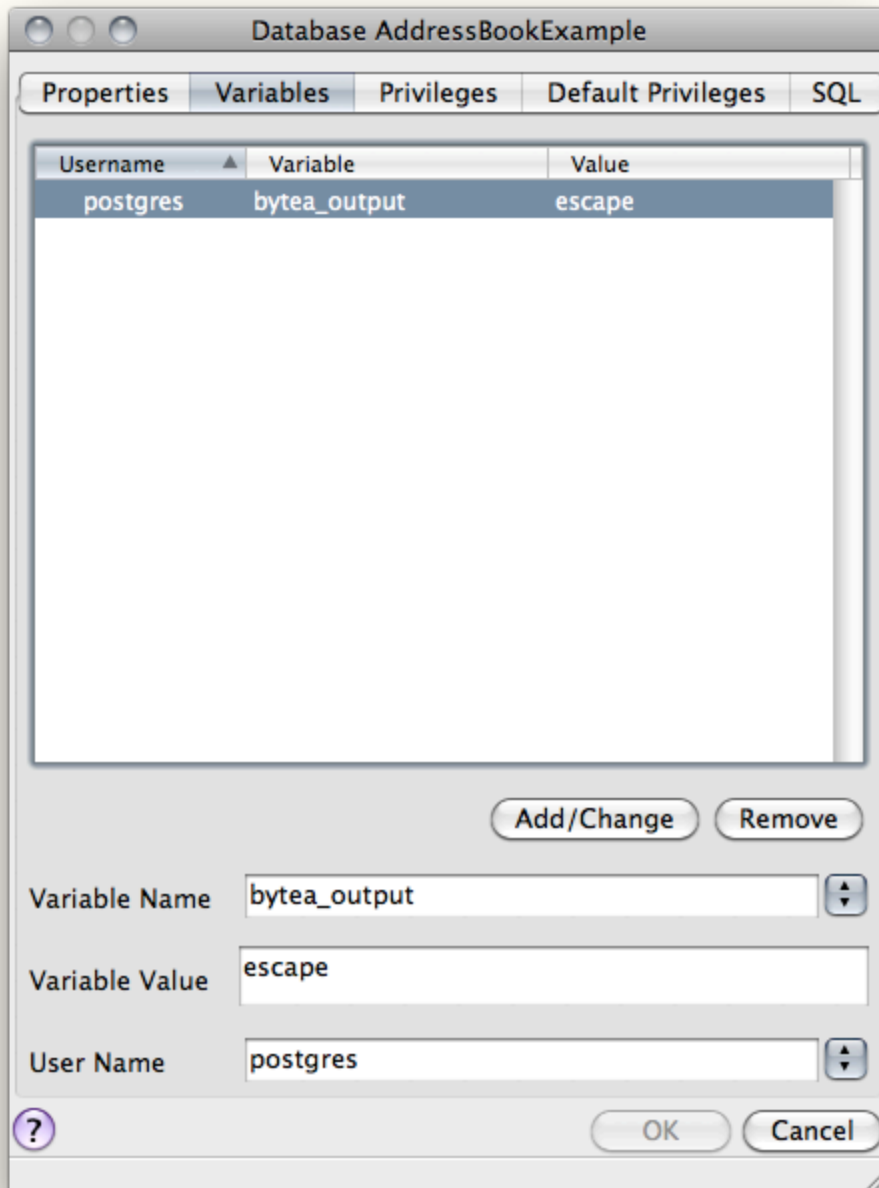
Note that the PostgreSQL JDBC driver will adjust timestamps handed to it by WebObjects into the local timezone of the machine running the JDBC driver. This will occur regardless of whether the database server was started up in UTC (by supplying `timezone=UTC` in `postgresql.conf`, or as a command line option). This also appears to occur regardless of whether the column type in use is `TIMESTAMP WITHOUT TIME ZONE` (which is the type used by the `ERPrototypes.framework` `dateTime` prototype), or `TIMESTAMP WITH TIME ZONE`. There are several workarounds:

1. Obviously if machine-local time is UTC, the problem is non-existent.
2. Pass the option `-Duser.timezone=UTC` to the JVM at application launch.
3. Have a static block in the application's `Application` class (or some superclass) that sets the Java default timezone:

```
static {
    TimeZone.setDefault(TimeZone.getTimeZone("UTC"));
}
```

Postgresql 9.0+ and bytea columns



In Postgresql 9.0, the default value for `bytea_output` changed from 'escape' to 'hex'. I've had an issue where the value written in to the db was read out incorrectly due to this. To fix the problem in PGAdmin3, right click on your database, and select the 'Properties' context menu option. In the properties window that pops up, select the 'Variables' tab, and set the value appropriately.





PostgresqlPlugin Framework

This framework is a plugin for the WebObjects JDBCAdaptor so that it is able to communicate with a PostgreSQL database server. The easiest way to get the `PostgresPlugin.framework` is to install [Project Wonder](#). Otherwise, you can download the built frameworks from the [WOComunity Jenkins server](#) (either [Wonder for WebObjects 5.3](#) or [Wonder of WebObjects 5.4](#)), extract the `PostgresqlPlugin.framework`, and place it in `~/Library/Frameworks`. You must add the `PostgresPlugin.framework` to your application's build path so that it is able to find the plugin.

Using PostgreSQL in Entity Modeler (WOLips)

Outline  PostgreSQL*  Default

Properties  Problems 

Config

Name	<input type="text" value="Default"/>
Prototype	<input type="text" value="EOJDBCPostgresqlPrototypes"/>
Adaptor	<input type="text" value="JDBC"/>
Username	<input type="text" value="whatever"/>
Password	<input type="text" value="whatever"/>
URL	<input type="text" value="jdbc:postgresql://yourhost/yourdatabase"/>
Driver	<input type="text" value="org.postgresql.Driver"/>
Plugin	<input type="text" value="PostgresqlPlugin"/>