

Installing JIRA on Oracle OC4J 10.1.3x

These instructions will help you install JIRA on OC4J 10.1.3.x. They have been tested on:

- 10.1.3.3.0
- 10.1.3.2
- 10.1.3.1.0

For older OC4J 10.1.x, 10.0.x and 9.x releases, please see the [older Orion/OC4J install docs](#).

1. 1. Unpack JIRA

Download and unzip the JIRA WAR (Webapp ARchive) distribution. A new directory containing JIRA will be created, hereafter referred to as `$JIRA_HOME`

2. 2. Database configuration

2.1. 2.1. Copy JDBC driver jar to OC4J

Download the JDBC driver .jar file for your database (see the list [here](#)), and copy it to OC4J's `j2ee/home/applib/` directory. For instance with PostgreSQL, one might have `j2ee/home/applib/pg73jdbc3.jar`.

2.2. 2.2 OC4J Database configuration

Edit `$OC4J/j2ee/home/config/data-sources.xml` and define a 'managed-data-source' and a 'connection-pool' for the database you wish JIRA to use. For example, using PostgreSQL:

```
<managed-data-source name="JiraDS" jndi-name="jdbc/JiraDS" connection-pool-name="JiraPool" />
<connection-pool name="JiraPool">
  <connection-factory factory-class="org.postgresql.Driver"
    user="jirauser"
    password="jirapassword"
    url="jdbc:postgresql://localhost:5432/jiradb" />
</connection-pool>
```

Database details (in bold) will vary depending on database - see [this page](#) for other database details.

Warning:

Oracle users note: the **setBigStringTryClob** parameter needs to be passed through to the JDBC driver for JIRA to be able to store very long strings in Oracle, but Orion and OC4J do not let this string through. We are not aware of a workaround for OC4J 10.1.3.x to pass this string through, and as such you may have problems running JIRA in OC4J 10.1.3.x on Oracle. See [JIRA-12564](#) for details. (For workarounds for other servers, including earlier versions of OC4J, see [Oracle 10g JDBC driver notes](#).)

2.3. 2.3. JIRA entityengine.xml configuration

Having installed a JDBC driver and configured a datasource, you now need to tell JIRA where to find the datasource and transaction manager, as well as what type of database you're using.

Open the `edit-webapp/WEB-INF/classes/entityengine.xml` file. There are two sections that must be changed.

First, near the top, locate:

```
<transaction-factory class="org.ofbiz.core.entity.transaction.JNDIFactory">
```

```
<user-transaction-jndi jndi-server-name="default" jndi-name="java:comp/env/UserTran
<transaction-manager-jndi jndi-server-name="default" jndi-name="java:comp/env/UserTran
</transaction-factory>
```

and remove the **/env**, so it reads:

```
<transaction-factory class="org.ofbiz.core.entity.transaction.JNDIFactory">
  <user-transaction-jndi jndi-server-name="default" jndi-name="java:comp/UserTran
  <transaction-manager-jndi jndi-server-name="default" jndi-name="java:comp/UserTran
</transaction-factory>
```

Second, at the bottom, customize the datasource section, specifying the right database type and jndi-name (highlighted in bold here):

```
<datasource name="defaultDS" field-type-name="postgres72"
  schema-name="public"
  helper-class="org.ofbiz.core.entity.GenericHelperDAO"
  check-on-start="true"
  use-foreign-keys="false"
  use-foreign-key-indices="false"
  check-fks-on-start="false"
  check-fk-indices-on-start="false"
  add-missing-on-start="true"
  check-indices-on-start="true">
  <jndi-jdbc jndi-server-name="default" jndi-name="jdbc/JiraDS" />
</datasource>
```

(In this example we're using PostgreSQL, and have added a schema-name="public" attribute as PostgreSQL requires.)

The jndi-name attribute in entityengine.xml must match the **ejb-location** attribute in your config/data-sources.xml file. Note the lack of **java:comp/env** in the jndi-name attribute.

3. 3. Build JIRA

Now build JIRA by typing **build** (Windows) or **./build.sh** (Unix) on the command line, in \$JIRA_HOME. This will produce the deployable WAR file in the \$JIRA_HOME/dist-generic directory. You can copy this elsewhere if you prefer.

4. 4. Deploy JIRA

Edit **j2ee/home/config/application.xml** to add the JIRA webapp to the default application like so:

```
<web-module id="jira" path="$JIRA_HOME/dist-generic/atlassian-jira-3.3.war" />
```

(where \$JIRA_HOME is the path to your JIRA distribution)

Now bind this "jira" webapp to a website. For example, to add JIRA to the default website edit **config/default-web-site.xml** and add the following line:

```
<web-app application="default" name="jira" root="/jira" />
```

Where:

- application="default" references the "default" application (application named "default" in application.xml)
- name="jira" references the id="jira" web-module defined in application.xml
- root="/jira" is the path off the website JIRA will be visible at, eg. <http://localhost:8888/jira/>.

5. 5. Start OC4J

First, edit the startup script, and set the OC4J_JVM_ARGS property. On Windows, edit

bin\oc4j.cmd and add the following line in bold:

```
set OC4J_JVM_ARGS=-Djavax.xml.transform.TransformerFactory=com.icl.saxon.TransformerFactoryImpl
set J2EE_HOME=%ORACLE_HOME%\j2ee\home
....
```

On Unix, edit bin/oc4j and add the following line in bold:

```
OC4J_JVM_ARGS=-Djavax.xml.transform.TransformerFactory=com.icl.saxon.TransformerFactoryImpl
J2EE_HOME=$ORACLE_HOME/j2ee/home
....
```

Note:

The -Djavax... parameter is to force OC4J to use the right XSLT engine - see [JIRA-8597](#).

Now download [saxon-noelfred-6.5.5.jar](#), the library the above parameter refers to, and copy it to j2ee/home/applib/.

Then start OC4J by going to the j2ee/home directory and running `java -jar oc4j.jar`. Watch the log/* files for any errors.

JIRA should become accessible at `http://localhost:8888/jira/`

6. Problems?

Here's a list of things to check:

1. Are there any errors on the console or j2ee/home/log/oc4j/log.xml and j2ee/home/log/* files? Usually if JIRA isn't starting, messages in the logs can get you on the right track.
2. Have you got a .jar file for saxon and for your database in **applib/**?
3. Is the JIRA webapp being deployed? Check in the j2ee/home/application-deployments/ directory - there should be a default/jira/ subdirectory if the webapp was deployed. If there isn't, check your server.xml and default-web-site files.
4. If OC4J seems to be using an old version of JIRA, delete the application-deployments/default/jira directory and restart. It is a good idea to do this anyway if you are unsure of the state of things.

If the problems persist, please [file a support request](#), attaching your logs and config files, and we'll take a look.

7. User-contributed notes

Have experiences to share with OC4J and JIRA? We welcome your thoughts. Please see the [user-contributed notes](#).