

# Installing JIRA on Orion Server / Oracle OC4J

These instructions will help you install JIRA on [OrionServer](#) and Oracle's App Server (OC4J).

This document is relevant to Orion 1.5.4, 1.6.0 and 2.0.x, and OC4J 9.0.4 up to 10.1.2.x.

## 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 Orion/OC4J

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

### 2.2. 2.2 Orion/OC4J Database configuration

Edit `$ORION/config/data-sources.xml` (or `$OC4J/j2ee/home/config/data-sources.xml`) and define a 'datasource' for the database you wish JIRA to use. For example, using PostgreSQL:

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="JIRA database"
  location="jdbc/JiraCoreDS"
  xa-location="jdbc/xa/JiraXADS"
  pooled-location="jdbc/JiraDS"
  connection-driver="org.postgresql.Driver"
  username="postgres"
  password="postgresPassword123"
  url="jdbc:postgresql:jiradb"
  inactivity-timeout="30"
/>
```

or for hsqldb (although you shouldn't use it in the long term, it has the virtue of requiring zero setup):

```
<data-source
  class="com.evermind.sql.DriverManagerDataSource"
  name="JIRA database"
  location="jdbc/JiraCoreDS"
  xa-location="jdbc/xa/JiraXADS"
  pooled-location="jdbc/JiraDS"
  connection-driver="org.hsqldb.jdbcDriver"
  username="sa"
  password=""
  url="jdbc:hsqldb:/tmp/jiradb"
  inactivity-timeout="30"
/>
```

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 work fully, but

Orion / OC4J do not let this string through. Please see the [Oracle 10g JDBC driver notes](#) for a workaround.

#### Note:

MySQL users note: If you are still using an 'ejb-location' in you datasource as advised in previous versions of this guide, please switch to using a 'pooled-location' instead. There is a problem with using the datasource named by 'ejb-location' when running against MySQL. See [JIRA-10132](#) for full details. In the case of running against MySQL the JNDI name you want to reference in your entityengine.xml is the name specified in the 'location' value (jdbc/JiraCoreDS, in the example above). The problem with running against this connection is that the datasource is not pooled and it could adversely effect performance. Please see the above mentioned issue for any updates regarding this configuration.

### 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/U
</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/UserTr
</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 Orion config/data-sources.xml file. Note the lack of **java:comp/env** in the jndi-name attribute.

### 3. App server modifications

**OC4J 9.0.4 users:** please also edit **j2ee/home/config/application.xml**, and comment out the **'../jsp/lib/taglib'** library:

```
<!--
  <library path="../jsp/lib/taglib"/>
-->
```

Otherwise you will get errors like this when importing data or upgrading:

```
2005-11-21 15:07:17,089 ERROR [atlassian.jira.upgrade.UpgradeManagerImpl] Exception thrown
java.lang.NoClassDefFoundError: org/dom4j/Element
    at org.jaxen.dom4j.Dom4jXPath.<init>(Dom4jXPath.java:103)
    at org.dom4j.xpath.DefaultXPath.parse(DefaultXPath.java:313)
    at org.dom4j.xpath.DefaultXPath.<init>(DefaultXPath.java:63)
    at org.dom4j.DocumentFactory.createXPath(DocumentFactory.java:188)
    at org.dom4j.DocumentHelper.createXPath(DocumentHelper.java:109)
    at com.atlassian.jira.upgrade.tasks.UpgradeTask_Build92.doUpgrade(UpgradeTask_Build
    ....
```

This does not apply to OC4J 10.0.2.x and above.

Finally, Orion does not come bundled with a compiler for JSP files. Hence Orion users (not OC4J) will need to copy tools.jar from the JDK lib directory to the Orion directory. This doesn't apply to OC4J.

### 4. 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.

### 5. 4. Deploy JIRA

Edit **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 Orion website edit **config/default-web-site.xml** (in OC4J, **config/http-web-site.xml**) and add the following line:

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

Where:

- application="default" references the "default" application (application.xml named "default" in server.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/.

### 6. 5. Start Orion with -userThreads property

Start Orion by using java -Djavax.xml.transform.TransformerFactory=com.icl.saxon.TransformerFactoryImpl -jar orion.jar -userThreads (oc4j.jar in j2ee/home for OC4J) in the directory where you installed Orion. Watch the log/\* files for any errors.

In Oracle App Server (OAS) -userThreads is set in the GUI screen. Go to <ContainerName> -> Administration -> Server Properties. On that screen, the field Command Line Options, OC4J Options has a default value of "-properties". It should be changed to **-properties -userThreads**

#### Note:

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

JIRA should become accessible at http://localhost/jira/ (using Orion, and assuming the default port 80 in default-web-site.xml) or http://localhost:8888/jira (OC4J).

**Warning:**

The `-userThreads` option is important -- without it, [services](#) will fail to run.

**7. Problems?**

Here's a list of things to check:

1. Are there any errors in the `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 your database in **lib/** (Orion) or **applib/**(OC4J)? In a console, try running `'jar tvf <jar>'` and make sure the `.jar` isn't corrupt, and contains the driver you expect (the 'connection-driver' in `config/data-sources.xml`; eg:

```
~/apps/oc4j-9.0.4/j2ee/home/applib$ jar tvf pg73jdbc3.jar | grep org.post
6229 Thu Aug 07 10:58:04 EST 2003 org/postgresql/Driver.class
```

3. Is the JIRA webapp being deployed? Check in the `application-deployments/` directory - there should be a `jira/` subdirectory if the webapp was deployed. If there isn't, check your `server.xml` and `default-web-site/http-web-site.xml` files.
4. Are you sure you referenced the **ejb-location** JNDI name (not **xa-location**) in `entityengine.xml`?
5. If Orion seems to be using an old version of your JIRA `.ear`, delete the `application-deployments/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.

**8. User-contributed notes**

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