

Installing JIRA on JBoss 4

These instructions will help you install JIRA on [JBoss 4](#) (with embedded Jetty or Tomcat). These instructions have been tested with JBoss 4.0.5GA, 4.0.2 and 4.0.1. JIRA 3.3 and above will **not work** on JBoss 4.0.0 due to its broken classloader - see [this issue](#). There are separate guides for [JBoss 3.x](#) and [JBoss 2.4](#).

If you're having problems, see [Getting Help](#).

1. 1. Configure JBoss

The application server (JBoss) is responsible for establishing a database connection, and making it available to webapps like JIRA as a "DataSource". DataSources are configured in XML files under JBoss' `server/default/deploy/` directory.

1.1. 1.1. Configure the datasource

JBoss 4.0.2+ comes with a usable (for JIRA 3.7+) [HSQLDB](#) datasource preconfigured. While HSQLDB isn't appropriate for production JIRA use, it is easy to get working and a good thing to try first. If you wish to try this HSQLDB datasource, please follow the steps below:

- If your version of JBoss is **4.0.5** to **4.2.0**, then no further modifications of JBoss config files are required, and you can skip to §2.
- If your version of JBoss is **prior to 4.0.5**, then you must replace the `/server/default/lib/hsqldb.jar` with `hsqldb-1.8.0.5.jar` to be able to use the preconfigured HSQLDB datasource.

For production use, you should configure JBoss to provide a datasource for an external database like PostgreSQL or MySQL. DataSources are configured in XML files under `server/default/deploy/`. Create a new file, `server/default/deploy/jira-ds.xml` containing:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DataSource for JIRA webapp,
      called 'JiraDS' (must be same in entityengine.xml) -->
<datasources>
  <local-tx-datasource>
    <jndi-name>JiraDS</jndi-name>
    <connection-url>jdbc:mysql://localhost/jirajboss?autoReconnect=true&useUnicode=
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>jira_mysql_username</user-name>
    <password>jira_mysql_password</password>
  </local-tx-datasource>
</datasources>
```

Here we have created a DataSource called 'JiraDS'. Change the connection-url and other details as required for your database (the [database configuration guides](#) show the correct connection-url and driver-class entry formats).

1.2. 1.2. Add the JDBC driver

Download the right JDBC driver for your database (again, see the relevant [database configuration guide](#) for where to get it). Copy the driver's jar file in JBoss' `servers/default/lib` directory. If you are using the built-in HSQLDB database, the JDBC driver (`hsqldb.jar`) is already present.

2. 2. Configuring the JIRA webapp

We assume you have the JIRA WAR/Webapp distribution.

2.1. 2.1. Modify transaction factory attributes

Open the `edit-webapp/WEB-INF/classes/entityengine.xml` file, and change:

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

to:

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

For JBoss 4.0.1 and 4.0.2, use **UserTransaction** rather than **ClientUserTransaction**.

2.2. 2.2. Specify database type and JNDI path

You now need to make changes to the `entityengine.xml` file to specify the database type and JNDI path. These changes will differ, depending on whether you are using **HSQldb** or an **external database** (not **HSQldb**). Go to the bottom of `edit-webapp/WEB-INF/classes/entityengine.xml` where the database type and `datasource` JNDI location are specified:

```
<datasource name="defaultDS" field-type-name="hsqldb"
  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="java:comp/env/jdbc/JiraDS"/>
```

For **HSQldb** (using the built-in `datasource` from §1.1) all you need to do is replace `java:comp/env/jdbc/JiraDS` with `java:/DefaultDS`

For **external databases** (not **HSQldb**):

- Change `field-type-name` appropriately for your database.
- If necessary, adjust `schema-name` appropriately your database.
- Replace `java:comp/env/jdbc/JiraDS` with `java:/JiraDS`

3. 3. Build JIRA

Now build a JIRA webapp by typing `build` (Windows) or `./build.sh` (Unix) on the command line, in the directory where you originally extracted JIRA to. This will produce a deployable WAR file in the `dist-generic` directory.

4. 4. Deploy the WAR in JBoss

Copy and rename `dist-generic/atlassian-jira-*.war` to `server/default/deploy/jira.war` in JBoss.

Stop and start JBoss to fully deploy the new JIRA webapp. Watch the startup logs for errors.

Note: in JBoss 4.0.2-4.0.5 you can expect to see one occurrence of a "Could not create an Appender" error, which is harmless - see [JIRA-6823](#). This is fixed in JBoss 4.2+. JBOSS 4.0.3SP1 is shipped with its own `log4j` which can return log errors as well. See [JIRA-14356](#) for details on how to address these errors.

Once JBoss has started, JIRA should be accessible at `http://localhost:8080/jira/` (change the host and port as needed; the 'jira' part is whatever the WAR name is).

5. User-contributed notes

Have experiences to share with JBoss and JIRA? We welcome your thoughts. Please see the [user-contributed JBoss 4 notes](#).