Connecting JIRA to HSQLDB

Note:

Before you begin: If you are already using JIRA, create an export of your data as an XML backup. You will then be able to transfer data from your old database to your new database, as described in Switching databases.

1. 1. Copy the HSQLDB driver to your application server

1. Download the HSQLDB JDBC driver - hsqldb-1.8.0.5.jar for JIRA 3.7+, or hsqldb-1.7.1-patched.jar for JIRA 3.6.5 and earlier. We **strongly recommend** upgrading to 3.7 if you wish to use hsqldb, as hsqldb 1.7.x is prone to data corruption.
2. Add the HSQLDB JDBC driver jar to the `common/lib/` directory.

2. 2. Configure your application server to connect to HSQLDB

1. Edit **`conf/server.xml`** (if you are using JIRA Standalone) and customise the `username`, `password`, `driverClassName`, `url`, `minEvictableIdleTimeMillis` and `timeBetweenEvictionRunsMillis` parameters for the Datasource. (If you are using JIRA WAR/EAR, edit the appropriate file on your application server; e.g. for Tomcat, edit **`conf/Catalina/localhost/jira.xml`**.)

```
<Server port="8005" shutdown="SHUTDOWN">

  <Service name="Catalina">

    <Connector port="8080"
      maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads=
      enableLookups="false" redirectPort="8443" acceptCount="100"
      connectionTimeout="20000" disableUploadTimeout="true" />

    <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">

        <Context path="" docBase="${catalina.home}/atlassian-jira" reloadable="false
          <Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
            username="sa"
            password=""
            driverClassName="org.hsqldb.jdbcDriver"
            url="jdbc:hsqldb:/path/to/jira/database/jiradb"
            minEvictableIdleTimeMillis="4000"
            timeBetweenEvictionRunsMillis="5000"
            maxActive="20"
            />

          <Resource name="UserTransaction" auth="Container" type="javax.transaction.
            factory="org.objectweb.jotm.UserTransactionFactory" jotm.timeout="60"/>
          <Manager className="org.apache.catalina.session.PersistentManager" saveOnR
        </Context>

      </Host>
    </Engine>
  </Service>
</Server>
```

(Note: if you can't find this section at all, you've probably got the wrong file - search for mentions of 'jira' in the files under `conf/`.)

Note:

In Tomcat 5.x, use '**`jdbc:hsqldb:${catalina.home}/database/jiradb`**' and '**`${catalina.home}`**' will be replaced with the JIRA path at runtime.

3. 4. Configure the JIRA Entity Engine

1. Edit **`atlassian-jira/WEB-INF/classes/entityengine.xml`** (if you are using JIRA Standalone) or **`edit-webapp/WEB-INF/classes/entityengine.xml`** (JIRA WAR/EAR), and change the `field-type-name` attribute to **`hsql`**. (If you forget to do this and start JIRA, it may create database tables incorrectly. See this page if this happens to you.)

```
<!-- DATASOURCE - You will need to update this tag for your installation.

-->
  <datasource name="defaultDS" field-type-name="hsql"
      helper-class="org.ofbiz.core.entity.GenericHelperDAO"
    check-on-start="true"
    use-foreign-keys="false"
    ...
```

Note:

If you are using JIRA WAR/EAR, your application server may require other changes to `entityengine.xml` (e.g. to customise the `jndi-jdbc` tag).

4. Next steps

You should now have an application server configured to connect to a database, and JIRA configured to use the correct database type. If you are using JIRA Standalone, start it up and watch the logs for any errors. If you are using the JIRA WAR/EAR distribution, rebuild and redeploy the webapp in your application server.

5. User-contributed notes

Have experiences to share with HSQL and JIRA? We welcome your thoughts. Please see the user-contributed HSQL notes.