

Connecting JIRA to DB2

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. Configure DB2

1. Create a database user which JIRA will connect as (e.g. **jirauser**).
2. Create a database for JIRA to store issues in (e.g. **jiradb**).
3. Ensure that the user has permission to connect to the database, and create and populate tables.

2. 2. Copy the DB2 driver to your application server

1. Add the DB2 JDBC driver jar (**db2jcc.jar**) to the `common/lib/` directory. (The DB2 JDBC driver is shipped with DB2.)

3. 3. Configure your application server to connect to DB2

1. Edit `conf/server.xml` (if you are using JIRA Standalone) and customise the `username`, `password`, `driverClassName` and `url` 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="[enter db username]"
            password="[enter db password]"
            driverClassName="com.ibm.db2.jcc.DB2Driver"
            url="jdbc:db2://localhost:50000/jiradb"
            [ delete the minEvictableIdleTimeMillis and timeBetweenEvictionRunsMilli
          />
          <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/`.)

2. If you are using JIRA Standalone, edit `conf/server.xml`, and delete the **`minEvictableIdleTimeMillis`** and **`timeBetweenEvictionRunsMillis`** attributes (which are only needed for HSQL, and degrade performance otherwise).

4. 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 `db2`. (If you forget to do this and start JIRA, it may create database tables incorrectly. See [this page](#) if this happens to you.)

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).

2. Set the value of the `schema-name` attribute to the name of the schema you are using for JIRA's database tables.

Here is an example of the `<datasource>` definition in `entitymodel.xml`:

```
<datasource name="defaultDS" field-type-name="db2"
  schema-name="DB2INST1"
  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"
  constraint-name-clip-length="15">
  <jndi-jdbc jndi-server-name="default" jndi-name="java:comp/env/jdbc/JiraDS">
</datasource>
```

Note:

Some versions of DB2 enforce a maximum length of 18 characters for a column name. However, the OFBiz entity-engine generates column names for primary keys based on the table name - so if the table name is longer than 15 characters, DB2 will not create the table. The solution is to modify the `entitymodel.xml` and add a `constraint-name-clip-length` attribute to the `<datasource>` tag in `entitymodel.xml`. Set the value of the `constraint-name-clip-length` attribute to `15`. This does *not* apply to DB2 on iSeries, AS/400 systems.

5. Additional notes

- At least some versions of DB2 require the length of CLOBs and BLOBs to be explicitly set in `WEB-INF/classes/entitydefs/fieldtype-db2.xml` (BLOB to `BLOB(1000)` and CLOB to `CLOB(2000)`). See [JIRA-4912](#).
- A user reports that DB2 on z/OS will not work with JIRA, as the SQL format differs. "UDB on LUW will automatically define index spaces etc, whereas on z/OS you define them via DDL."

6. 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.

7. User-contributed notes

Have experiences to share with DB2 and JIRA? We welcome your thoughts. Please see the [user-contributed DB2 notes](#).