

Connecting JIRA to MySQL

This document applies to MySQL 3.53.x up to 5.0.x.

Note: A [Linux-specific version of these instructions](#) is available.

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 MySQL

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**).

The database must have a character set of UTF8. Enter the following command from within the MySQL command client:

```
create database jiradb character set utf8;
```

(if you want your database to be named **jiradb**).

3. Ensure that the user has permission to connect to the database, and create and populate tables.

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

1. Download the MySQL [Connector/J JDBC driver](#), eg. version 3.1.12 ([tar.gz](#) or [zip](#)) which we have verified to work. NOTE: the older 3.1.11 driver is [broken](#).

Note:

A user has reported encountering problems using the Resin JDBC driver for MySQL. However, the Connector/J driver from MySQL works correctly ([except for version 3.1.11](#)).

2. Add the MySQL JDBC driver jar (**mysql-connector-java-3.x.x-bin.jar**) to the `common/lib/` directory. NOTE: Do not place the Debug Driver (`mysql-connector-java-3.x.x-bin-g.jar`) on the CLASSPATH as this can cause issues ([JIRA-8674](#)).

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

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`.)

Warning:

The URL in the XML below assumes a UTF-8 database - i.e. created with `create database jiradb character set utf8`. If you don't specify `character set utf8` you risk getting 'Data truncation: Data too long for column' errors when importing data or corruption of non-supported characters. See [storing non-ASCII characters in MySQL](#) for details.

Note: if entered into an XML file, escape the '&' with '&' as follows:

```
<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.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost/jiradb?autoReconnect=true&useUnicode=true
          [ 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/.)

The URL above assumes a UTF-8 database - ie. created with `create database jiradb character set utf8;`. If you don't specify character set `utf8` you risk getting 'Data truncation: Data too long for column' errors when importing data. See [storing non-ASCII characters in MySQL](#) for details.

Note:

MySQL closes idle connection after 8 hours, so the `autoReconnect=true` is necessary to tell the driver to reconnect

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 `mysql`. (If you forget to do this and start JIRA, it may create database tables incorrectly. See [this page](#) if this happens to you.) Also delete the `schema-name="PUBLIC"` attribute, if it exists:

```

<!-- DATASOURCE - You will need to update this tag for your installation.
-->
<datasource name="defaultDS" field-type-name="mysql"
  [ delete this, if it exists: schema-name="PUBLIC" ]
  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 customize the `jndi-jdbc` tag).

5. 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.

6. User-contributed notes

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