

Connecting JIRA to Oracle

This document applies to Oracle 9i and 10g. (Since Oracle Corporation is dropping support for Oracle 8i it has become increasingly difficult for Atlassian to support JIRA installations using this version of Oracle database. See [this link](#) for further information on Oracle's obsolescence/desupport of 8i.)

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 Oracle

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 Oracle driver to your application server

WARNING: please make sure you get the **10.1.0.5** version of the driver. Every other version has problems:

- The 9i drivers don't support CLOBs, so are unusable.
- The 10g Release 2 JDBC driver (10.2.0.1.0) hangs with some databases.
- The 10g Release 1 JDBC driver (10.1.0.4) does not hang, but [throws ArrayIndexOutOfBoundsException](#). A second user reports that it silently fails to import workflows in Oracle 9i, and JIRA later dies with a NullPointerException.
- The latest 10.1.0.5 driver allegedly fixes the ArrayIndexOutOfBoundsException, and we have at least one report of it working without problems.

Anecdotally, these problems seem to affect 9i users more than 10g. Thus:

- If you are using **Oracle 9i**, we recommend you avoid Oracle's drivers altogether, and buy the [I-net software JDBC driver](#), which is known to work without problems. Try the 10g driver at your own risk.
- If you are using **Oracle 10g**, download the 10.1.0.5 driver from [Oracle's site](#).

Each of the problems listed above has been discovered at great cost to our users, and support cases have sometimes dragged for months. We thus **strongly recommend avoiding Oracle** if you have any choice in the matter, for the benefit of all concerned.

1. Download the Oracle JDBC driver from [I-net software](#), or [Oracle's site \(direct link\)](#). If you are using Oracle 9i please use Oracle 10g JDBC drivers as they are required for CLOB support.
2. Add the Oracle JDBC driver jar (**ojdbc14.jar**) to the `common/lib/` directory (for Tomcat), or the relevant `lib` directory in your app server..

Note:

Both Oracle 9 and 10 drivers are called *ojdbc14.jar*, so don't rely on the name, or assume that an existing driver present in your directory is the correct one. The correct 10g R1 driver is 1,378,346 bytes, whereas the wrong 9i driver is 1.14Mb. The (broken) 10gR2 driver is 1,536,979 bytes.

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

3.1. Tomcat (or JIRA Standalone)

In an editor, open **conf/server.xml** (JIRA Standalone) or **conf/Catalina/localhost/jira.xml** (regular Tomcat). Locate the section:

```
<Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
  username="sa"
  password=""
  driverClassName="org.hsqldb.jdbcDriver"
  url="jdbc:hsqldb:${catalina.home}/database/jiradb"
  minEvictableIdleTimeMillis="4000"
  timeBetweenEvictionRunsMillis="5000"
  maxActive="20" />
```

(Note: if you can't find a section like this at all, you've probably got the wrong file. Search for mentions of 'Jira' in the files under **conf/**.)

Replace this section with one of the following, depending on the driver you're using.

3.1.1. I-Net JDBC driver

```
<Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
  username="[enter db username]"
  password="[enter db password]"
  driverClassName="com.inet.ora.OraDriver"
  url="jdbc:inetora:localhost:1521:jiradb"
  connectionProperties="streamstolob=true"
  maxActive="20" />
```

3.1.2. Oracle 10 JDBC driver

```
<Resource name="jdbc/JiraDS" auth="Container" type="javax.sql.DataSource"
  username="[enter db username]"
  password="[enter db password]"
  driverClassName="oracle.jdbc.driver.OracleDriver"
  url="jdbc:oracle:thin:@localhost:1521:jiradb"
  connectionProperties="SetBigStringTryClob=true"
  maxActive="20" />
```

Customise the username, password, database server hostname (assumed to be localhost above, and database name (jiradb above).

Please ensure you have the **connectionProperties** setting, which is needed to store unlimited-text fields in Oracle. Without this, Oracle will only store strings up to 32K bytes in size.

If you were previously using **hsqldb** make sure you have *removed* have the **minEvictableIdleTimeMillis** and **timeBetweenEvictionRunsMillis** attributes. They will slow JIRA down if present.

3.2. JIRA WAR/EAR distribution

If you're deploying JIRA on another application server, you'll need to build a JIRA Webapp from the WAR/EAR distribution. Refer primarily to the [application server setup guides](#) for details of configuring your application server.

There is one Oracle-specific tweak required - the **SetBigStringTryClob** (Oracle) or **streamstolob** (I-Net) connection property must be set. Please see the [following document](#) for how to set this.

4. 4. Configure the JIRA Entity Engine

- 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 **oracle10g**. (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, or replace the schema name with the schema you are using (this makes JIRA start faster too - see [this thread](#)):

```
<!-- DATASOURCE - You will need to update this tag for your installation.
-->
<datasource name="defaultDS" field-type-name="oracle10g"
  helper-class="org.ofbiz.core.entity.GenericHelperDAO"
  check-on-start="true"
  use-foreign-keys="false"
  ...
```

The `field-type-name` **must** be set to `oracle10g` (**not** `oracle`) for this configuration to work.

- 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). Please refer to the [documentation for your application server](#) for details.

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. Troubleshooting

6.1. Importing into Oracle from another database

If you have exported XML data from a JIRA instance running a different database than Oracle, and then the import into Oracle fails, you may be affected by [JIRA-14085](#).

7. User-contributed notes

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