Connecting JIRA to Sybase

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 Sybase

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.

Note:

By default Sybase does not allow `NULL` values for a table's columns - which can be problematic for JIRA.
A solution to this issue is to setup the Sybase database used by JIRA to allow `NULL` values by default, by setting the `ALLOW DEFAULTS ON` option. This procedure is described at:
http://manuals.sybase.com/onlinebooks/group-as/asg1250e/svrtsg/@Generic__BookTextView/15380. See also JRA-4815.

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

1. Download the Sybase JDBC driver from http://jtds.sourceforge.net/
2. Add the Sybase JDBC driver jar (**`jtds-1.2.jar`**) to the `common/lib/` directory.

Note:

An offical Sybase driver, called jConnect, is available. However it seems not to be able to store CLOBs larger than 16kb (see JRA-6679), and so we recommend jTDS instead

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

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="net.sourceforge.jtds.jdbc.Driver"

            url="jdbc:jtds:sybase://yourmachinename:portnumber/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 **`sybase`**. (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:

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

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

   - 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).
   - A user of JBoss and Sybase ASA 9.0.2 reported getting errors (e.g. `com.sybase.jdbc2.jdbc.SybSQLException: ASA Error -195: Column 'PASSWORD_HASH' in table 'userbase' cannot be NULL`) even after nulls were allowed (`select * from systable where "option" like '%null%'`). This was eventually resolved by modifying the JBoss datasource definition to include: `<new-connection-sql> sp_dboption jira, "allow nulls by default", true </new-connection-sql>`

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 Sybase and JIRA? We welcome your thoughts. Please see the user-contributed Sybase notes.