JIRA

How to use Jetty with JIRA

These instructions have been tested against:

Jetty Version	Jira Version	Tx Mgr & Datasource	JVM	Servlet & JSP
6.0.1	3.6.5	JOTM 2.1.10 & HSQL 1.7	1.5	2.5 & 2.1
6.1	3.6.5	JOTM 2.1.10 & HSQL 1.7	1.5	2.5 & 2.1
6.1	3.6.5	Atomikos 3.1.0 & HSQL 1.7	1.5	2.5 & 2.1
6.1	3.6.5	Atomikos 3.1.0 & HSQL 1.7	1.4	2.5 & 2.0
6.1.1	3.8.1	JOTM 2.1.10 & HSQL 1.8.0.5	1.6	2.5 & 2.0
6.1.9	3.12.1	Atomikos 3.2.3 & HSQL 1.8.0.7	1.5.0_11	2.5 & 2.1

Prepare the JIRA webapp

1. if you've downloaded the standalone distribution of JIRA:
    a. copy the commons/lib/hsqldb-*.jar into $jetty.home/lib/ext
    b. copy the atlassian-jira directory into your $jetty.home/webapps-plus directory (create `webapps-plus` if it doesn't exist, e.g. on 6.0).
    c. edit $jetty.home/webapps-plus/atlassian-jira/WEB-INF/classes/entityengine.xml to change from the tomcat 5-specific setup for the JNDI binding for the UserTransaction. It must be java:comp/UserTransaction and NOT java:comp/env/UserTransaction.
2. if you've downloaded the webapp distribution of JIRA:
    a. download the hsqldb jar from the JIRA site (if you're not using the latest JIRA, go here and find the right docs version) and copy to $jetty.home/lib/ext
    b. without further configuring anything, build the webapp with `build.sh`/`build.bat`, following the readme.txt instructions.
    c. make the directory $jetty.home/webapps-plus/atlassian-jira.
    d. expand the atlassian-jira-3.6.5.war into it.
    e. edit $jetty.home/webapps-plus/atlassian-jira/WEB-INF/classes/entityengine.xml to ensure that the JNDI binding for the UserTransaction is java:comp/UserTransaction. Also check that if you are using hsqldb that the `<datasource>` element at the bottom of the file has attribute `field-type-name="hsql"`.
3. regardless of which distribution you downloaded, now edit the $jetty.home/webapps-plus/atlassian-jira/WEB-INF/web.xml file:
    a. add a <resource-ref> for the DataSource used by JIRA:

```
<resource-ref>
        <description>Database for JIRA</description>
        <res-ref-name>jdbc/JiraDS</res-ref-name>
        <res-type>javax.sql.DataSource</res-type>
        <res-auth>Container</res-auth>
</resource-ref>
```

b. if you want to use a mail Session JNDI lookup, then also add a <resource-ref> for it too:

```
<resource-ref>
    <res-ref-name>mail/Session</res-ref-name>
    <res-type>javax.mail.Session</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
```

> It is important that you copy the jira webapp into the directory that is setup to deploy webapps using JNDI. If you are using the default jetty setup from the distribution, then this directory will be $jetty.home/webapps-plus **not** $jetty.home/webapps.

Prepare Jetty

1. set up the jetty installation for the transaction manager of your choice, following the instructions for JOTM here or Atomikos here (Atomikos recommended).
2. copy $jetty.home/lib/jsp-2.0/jcl104-over-slf4j-1.0.1.jar and $jetty.home/lib/slf4j-simple-1.0.1.jar to $jetty.home/lib/. This configures your commons-logging implementation so that JIRA (and JOTM if you are using it) send their log messages to the same log as the jetty server, using SLF4J. Alternatively if you want to use commons-logging, you will need to copy the commons-logging jar and a commons logging impl into the $jetty.home/lib/ directory instead.
3. setup the JNDI resources. To save you some time, you can use the jetty-jira-jndi.xml file attached to this page, copy it into $jetty.home/etc, and uncomment either the Atomikos or the JOTM setup as you prefer. The file also contains a mail Session JNDI binding which you can uncomment and configure if you elected to setup a <resource-ref> for it in the JIRA web.xml in previous steps.
4. copy jta.properties and place it on $jetty.home/resources

There are 2 ways to deploy:

Using WebAppDeployer

1. Configuration
   On your jetty-plus.xml, uncomment(if its commented out) the ff:

```
<Call name="addLifeCycle">
      <Arg>
        <New class="org.mortbay.jetty.deployer.WebAppDeployer">
          <Set name="contexts"><Ref id="Contexts"/></Set>
          <Set name="webAppDir"><SystemProperty name="jetty.home"
default="."/>/webapps-plus</Set>
          <Set name="parentLoaderPriority">false</Set>
          <Set name="extract">true</Set>
          <Set name="allowDuplicates">false</Set>
          <Set name="defaultsDescriptor"><SystemProperty name="jetty.home"
default="."/>/etc/webdefault.xml</Set>
          <Set name="configurationClasses"><Ref id="plusConfig"/></Set>
        </New>
      </Arg>
    </Call>
```

2. Running:

```
java -jar start.jar etc/jetty.xml etc/jetty-plus.xml
etc/jetty-jira-jndi.xml
```

✅ **Shortening the runline**

You can move all of the jetty xml configuration into a single file if you want to shorten the runline. Better yet, put it all in etc/jetty.xml and you won't even have to specify any config file on the runline, as it is the default:

```
java -jar start.jar
```

Using ContextDeployer

1. Configuration:
   Create a "jira-context.xml" in $jetty.home/contexts and append the ff:

```
<?xml version="1.0"  encoding="ISO-8859-1"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://jetty.mortbay.org/configure.dtd">
<Configure class="org.mortbay.jetty.webapp.WebAppContext">

  <Set name="contextPath">/</Set>

  <Set name="war"><SystemProperty name="jetty.home"
default="."/>/webapps-plus/atlassian-jira</Set>

  <Set name="defaultsDescriptor"><SystemProperty name="jetty.home"
default="."/>/etc/webdefault.xml</Set>

  <!--  NEEDED WHEN USING JNDI -->
  <Set name="configurationClasses">
  <Array id="plusConfig" type="java.lang.String">
    <Item>org.mortbay.jetty.webapp.WebInfConfiguration</Item>
    <Item>org.mortbay.jetty.plus.webapp.EnvConfiguration</Item>
    <Item>org.mortbay.jetty.plus.webapp.Configuration</Item>
    <Item>org.mortbay.jetty.webapp.JettyWebXmlConfiguration</Item>
    <Item>org.mortbay.jetty.webapp.TagLibConfiguration</Item>
  </Array>
  </Set>

</Configure>
```

2. Running:

```
java -jar start.jar etc/jetty.xml etc/jetty-jira-jndi.xml
```

Addendum

For the curious, here's what the jetty-jira.xml file looks like:

```
<!--
=================================================
==================== -->
<!-- Configuring Transactions and XA
Datasources                          -->
<!-- Uncomment EITHER the setup for Atomikos
OR for Jotm                 -->
<!--
```

```
=============================================
=================== -->
<!-- START OF ATOMIKOS
  <Call class="java.lang.System"
name="setProperty">
    <Arg>com.atomikos.icatch.file</Arg>
    <Arg><SystemProperty name="jetty.home"
default="."
/>/resources/jta.properties</Arg>
  </Call>
  <New id="tx"
class="org.mortbay.jetty.plus.naming.Transac
tion">
   <Arg>
     <New
class="com.atomikos.icatch.jta.UserTransacti
onImp"/>
   </Arg>
  </New>


   <New id="jira-ds"
class="org.mortbay.jetty.plus.naming.Resourc
e">
      <Arg>jdbc/JiraDS</Arg>
      <Arg>
       <New
class="com.atomikos.jdbc.nonxa.NonXADataSour
ceBean">
         <Set
name="Url">jdbc:hsqldb:./database/jiradb</Se
```

```
t>
         <Set
name="driverClassName">org.hsqldb.jdbcDriver
</Set>
         <Set name="user">sa</Set>
       </New>
      </Arg>
    </New>
  END OF ATOMIKOS -->
  <!-- START OF JOTM  CONFIG
  <New id="jotm"
class="org.objectweb.jotm.Jotm">
    <Arg type="boolean">True</Arg>
    <Arg type="boolean">False</Arg>
    <Call id="ut"
name="getUserTransaction"/>
    <Call id="tm"
name="getTransactionManager"/>
  </New>
  <Call name="setAttribute">
    <Arg>txmgr</Arg>
    <Arg><Ref id="tm"/></Arg>
  </Call>

  <New
class="org.mortbay.jetty.plus.naming.Resourc
e">

<Arg>javax.transaction.TransactionManager</A
rg>
    <Arg><Ref id="ut"/></Arg>
```

```
  </New>

  <New id="tx"
class="org.mortbay.jetty.plus.naming.Transac
tion">
   <Arg>
     <Ref id="ut"/>
   </Arg>
  </New>

  <New
class="org.mortbay.jetty.plus.naming.Resourc
e">
    <Arg>XAJiraDS</Arg>
    <Arg>
      <New id="myxadatasource"
class="org.enhydra.jdbc.standard.StandardXAD
ataSource">
        <Set
name="DriverName">org.hsqldb.jdbcDriver</Set
>
        <Set
name="Url">jdbc:hsqldb:./database/jiradb</Se
t>
        <Set name="User">sa</Set>
        <Set name="Password"></Set>
        <Set name="transactionManager"><Ref
id="tm"/></Set>
      </New>
    </Arg>
  </New>
```

```
  <New id="mydatasource"
class="org.mortbay.jetty.plus.naming.Resourc
e">
    <Arg>jdbc/JiraDS</Arg>
    <Arg>
      <New
class="org.enhydra.jdbc.pool.StandardXAPoolD
ataSource">
        <Arg><Ref
id="myxadatasource"/></Arg>
        <Set
name="DataSourceName">XAJiraDS</Set>
        <Set name="User">sa</Set>
      </New>
    </Arg>
  </New>
 END OF JOTM -->

  <!--
=================================================
==================== -->
  <!-- Optional Mail Session setup
-->
  <!--
=================================================
==================== -->
  <New id="mail"
class="org.mortbay.jetty.plus.naming.Resourc
e">
     <Arg>mail/Session</Arg>
```

```
      <Arg>
        <New
class="org.mortbay.naming.factories.MailSess
ionReference">
          <Set name="user">CHANGEME</Set>
          <Set name="password">CHANGEME</Set>
          <Set name="properties">
            <New
class="java.util.Properties">
              <Put
name="mail.smtp.host">CHANGEME</Put>
              <Put
name="mail.from">CHANGEME</Put>
              <Put
name="mail.debug">true</Put>
            </New>
           </Set>
```

```
        </New>
      </Arg>
</New>
```

At least two people have independently experienced NullPointerExceptions from XAPool with this configuration. See JRA-12653 for details.

- If you are one of them, can you please raise a JIRA in the Jetty jira with details of the versions that you used - specifically which TM. Atomikos appears to be the better maintained TM at the moment, so it is recommended for XA.