

Quartz Resource Adapter usage

Example quartz-app can be browsed at <https://github.com/apache/tomee/tree/master/examples/quartz-app>

Note this example is somewhat dated. It predates the schedule API which was added to EJB 3.1. Modern applications should use the schedule API which has many, if not all, the same features as Quartz. In fact, Quartz is the engine that drives the `@Schedule` and `ScheduleExpression` support in OpenEJB and TomEE.

Despite being dated from a scheduling perspective it is still an excellent reference for how to plug-in and test a custom Java EE Resource Adapter.

Project structure

As `.rar` files do not do well on a standard classpath structure the goal is to effectively "unwrap" the `.rar` so that its dependencies are on the classpath and its `ra.xml` file can be found in scanned by OpenEJB.

We do this by creating a mini maven module to represent the rar in maven terms. The `pom.xml` of the "rar module" declares all of the jars that would be inside `.rar` as maven dependencies. The `ra.xml` file is added to the project in `src/main/resources/META-INF/ra.xml` where it will be visible to other modules.

```
quartz-app
quartz-app/pom.xml
quartz-app/quartz-beans
quartz-app/quartz-beans/pom.xml
quartz-app/quartz-beans/src/main/java/org/superbiz/quartz/JobBean.java
quartz-app/quartz-beans/src/main/java/org/superbiz/quartz/JobScheduler.java
quartz-app/quartz-beans/src/main/java/org/superbiz/quartz/QuartzMdb.java
quartz-app/quartz-beans/src/main/resources/META-INF
quartz-app/quartz-beans/src/main/resources/META-INF/ejb-jar.xml
quartz-app/quartz-beans/src/test/java/org/superbiz/quartz/QuartzMdbTest.java
quartz-app/quartz-ra
quartz-app/quartz-ra/pom.xml
quartz-app/quartz-ra/src/main/resources/META-INF
quartz-app/quartz-ra/src/main/resources/META-INF/ra.xml
```

ra.xml

The connector in question has both inbound and outbound Resource Adapters. The inbound Resource Adapter can be used to drive message driven beans (MDBs)

the outbound Resource Adapter, `QuartzResourceAdapter`, can be injected into any component via `@Resource` and used to originate and send messages or events.

```

<connector xmlns="http://java.sun.com/xml/ns/j2ee"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
           http://java.sun.com/xml/ns/j2ee/connector_1_5.xsd"
           version="1.5">

  <description>Quartz ResourceAdapter</description>
  <display-name>Quartz ResourceAdapter</display-name>

  <vendor-name>OpenEJB</vendor-name>
  <eis-type>Quartz Adapter</eis-type>
  <resourceadapter-version>1.0</resourceadapter-version>

  <resourceadapter id="QuartzResourceAdapter">
    <resourceadapter-class>
org.apache.openejb.resource.quartz.QuartzResourceAdapter</resourceadapter-class>

    <inbound-resourceadapter>
      <messageadapter>
        <messagelistener>
          <messagelistener-type>org.quartz.Job</messagelistener-type>
          <activationspec>
            <activationspec-class>
org.apache.openejb.resource.quartz.JobSpec</activationspec-class>
          </activationspec>
        </messagelistener>
      </messageadapter>
    </inbound-resourceadapter>

  </resourceadapter>
</connector>

```

Using the Outbound Resource Adapter

Here we see the outbound resource adapter used in a stateless session bean to schedule a job that will be executed by the MDB

```

package org.superbiz.quartz;

import org.apache.openejb.resource.quartz.QuartzResourceAdapter;
import org.quartz.Job;
import org.quartz.JobDetail;
import org.quartz.JobExecutionContext;
import org.quartz.JobExecutionException;
import org.quartz.Scheduler;
import org.quartz.SimpleTrigger;

import javax.ejb.Stateless;
import javax.naming.InitialContext;
import java.util.Date;

@Stateless
public class JobBean implements JobScheduler {

    @Override
    public Date createJob() throws Exception {

        final QuartzResourceAdapter ra = (QuartzResourceAdapter) new InitialContext()
            .lookup("java:openejb/Resource/QuartzResourceAdapter");
        final Scheduler s = ra.getScheduler();

        //Add a job type
        final JobDetail jd = new JobDetail("job1", "group1", JobBean.MyTestJob.class);
        jd.getJobDataMap().put("MyJobKey", "MyJobValue");

        //Schedule my 'test' job to run now
        final SimpleTrigger trigger = new SimpleTrigger("trigger1", "group1", new
Date());
        return s.scheduleJob(jd, trigger);
    }

    public static class MyTestJob implements Job {

        @Override
        public void execute(JobExecutionContext context) throws JobExecutionException
        {
            System.out.println("This is a simple test job to get: " + context
                .getJobDetail().getJobDataMap().get("MyJobKey"));
        }
    }
}

```

Receiving data from the Inbound Resource Adapter

```
package org.superbiz.quartz;

import org.quartz.Job;
import org.quartz.JobExecutionContext;
import org.quartz.JobExecutionException;

import javax.ejb.ActivationConfigProperty;
import javax.ejb.MessageDriven;

@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "cronExpression", propertyValue = "* * * * ?")
})
public class QuartzMdb implements Job {

    @Override
    public void execute(JobExecutionContext jobExecutionContext) throws
    JobExecutionException {
        System.out.println("Executing Job");
    }
}
```

Test case

```
package org.superbiz.quartz;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;

import javax.naming.Context;
import javax.naming.InitialContext;
import java.util.Date;
import java.util.Properties;

public class QuartzMdbTest {

    private static InitialContext initialContext = null;

    @BeforeClass
    public static void beforeClass() throws Exception {

        if (null == initialContext) {
```

```

        Properties properties = new Properties();
        properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
"org.apache.openejb.core.LocalInitialContextFactory");

        initialContext = new InitialContext(properties);
    }
}

@AfterClass
public static void afterClass() throws Exception {
    if (null != initialContext) {
        initialContext.close();
        initialContext = null;
    }
}

@Test
public void testLookup() throws Exception {

    final JobScheduler jbi = (JobScheduler) initialContext.lookup("JobBeanLocal");
    final Date d = jbi.createJob();
    Thread.sleep(500);
    System.out.println("Scheduled test job should have run at: " + d.toString());
}

@Test
public void testMdb() throws Exception {
    // Sleep 3 seconds and give quartz a chance to execute our MDB
    Thread.sleep(3000);
}
}

```

Running

```

-----
T E S T S
-----

Running org.superbiz.quartz.QuartzMdbTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/quartz-app/quartz-beans
INFO - openejb.base = /Users/dblevins/examples/quartz-app/quartz-beans
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Found ConnectorModule in classpath: /Users/dblevins/examples/quartz-app/quartz-
ra/target/quartz-ra-1.0.jar

```

```
INFO - Found EjbModule in classpath: /Users/dblevins/examples/quartz-app/quartz-
beans/target/classes
INFO - Beginning load: /Users/dblevins/examples/quartz-app/quartz-ra/target/quartz-ra-
1.0.jar
INFO - Extracting jar: /Users/dblevins/examples/quartz-app/quartz-ra/target/quartz-ra-
1.0.jar
INFO - Extracted path: /Users/dblevins/examples/quartz-app/quartz-ra/target/quartz-ra-
1.0
INFO - Beginning load: /Users/dblevins/examples/quartz-app/quartz-beans/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/quartz-app/quartz-
beans/classpath.ear
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean JobBean: Container(type=STATELESS,
id=Default Stateless Container)
INFO - Configuring Service(id=QuartzResourceAdapter, type=Resource, provider-
id=QuartzResourceAdapter)
INFO - Configuring Service(id=quartz-ra-1.0, type=Container, provider-id=Default MDB
Container)
INFO - Enterprise application "/Users/dblevins/examples/quartz-app/quartz-
beans/classpath.ear" loaded.
INFO - Assembling app: /Users/dblevins/examples/quartz-app/quartz-beans/classpath.ear
INFO - Jndi(name=JobBeanLocal) --> Ejb(deployment-id=JobBean)
INFO - Jndi(name=global/classpath.ear/quartz-
beans/JobBean!org.superbiz.quartz.JobScheduler) --> Ejb(deployment-id=JobBean)
INFO - Jndi(name=global/classpath.ear/quartz-beans/JobBean) --> Ejb(deployment-
id=JobBean)
INFO - Created Ejb(deployment-id=JobBean, ejb-name=JobBean, container=Default
Stateless Container)
INFO - Created Ejb(deployment-id=QuartzMdb, ejb-name=QuartzMdb, container=quartz-ra-
1.0)
Executing Job
INFO - Started Ejb(deployment-id=JobBean, ejb-name=JobBean, container=Default
Stateless Container)
INFO - Started Ejb(deployment-id=QuartzMdb, ejb-name=QuartzMdb, container=quartz-ra-
1.0)
INFO - Deployed Application(path=/Users/dblevins/examples/quartz-app/quartz-
beans/classpath.ear)
This is a simple test job to get: MyJobValue
Scheduled test job should have run at: Fri Oct 28 17:05:12 PDT 2011
Executing Job
Executing Job
Executing Job
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.971 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
```