

Description:

=====

Integrating JBoss EAP 6.2 with Tibco EMS.

Resolution:

=====

JBoss EAP 6.2 has provided Generic JMS Resource Adapter to integrate with a Third-party JMS Provider. You can use the following steps to configure a Generic JMS Resource Adapter within JBoss EAP 6.2 to work with EMS.

Configure a Generic JMS Resource Adapter for EMS within JBOSS EAP 6.2:

1). Create an ObjectFactory implementation for the JNDI binding of queues and topics:

1.1). Using the code below as a template, replace the server details with your JMS provider server values.

=====

```
import java.util.Hashtable;
import java.util.Properties;
```

```
public class RemoteJMSObjectFactory implements ObjectFactory {
```

```
    private Context context = null;
```

```
    public RemoteJMSObjectFactory() {
    }
}
```

```
    public Object getObjectInstance(Object obj, Name name, Context nameCtx,
        Hashtable<?, ?> environment) throws Exception {
```

```
        try {
            String jndi = (String) obj;
```

```
            final Properties env = new Properties();
            env.put(Context.INITIAL_CONTEXT_FACTORY,
                "com.tibco.tibjms.naming.TibjmsInitialContextFactory");
            env.put(Context.URL_PKG_PREFIXES, "com.tibco.tibjms.naming");
            env.put(Context.PROVIDER_URL, "tcp://TIBCO_HOST:TIBCO_PORT");
```

```
            context = new InitialContext(env);
            Object o = context.lookup(jndi);
```

```

        return o;
    } catch (NamingException e) {
        e.printStackTrace();
        throw e;
    }
}
}
}
=====

```

1.2). Compile the above code and save the resulting class file in a JAR file named remoteJMSObjectFactory.jar.

2). Create a genericjms module for your JBoss EAP 6 instance:

2.1). Create the following directory structure:

EAP\_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main

2.2). Copy the remoteJMSObjectFactory.jar file to

EAP\_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main

2.3). Copy the binaries required for the provider's JMS implementation to

EAP\_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main. For Tibco EMS, the binaries required are tibjms.jar and tibcrypt.jar from the Tibco installation's /lib directory.

2.4). Create a module.xml file in

EAP\_HOME/modules/system/layers/base/org/jboss/genericjms/provider/main as below, listing the JAR files from the previous steps as resources:

```

<module xmlns="urn:jboss:module:1.1" name="org.jboss.genericjms.provider">
  <resources>
    <resource-root path="tibjms.jar"/>
    <resource-root path="tibcrypt.jar"/>
    <resource-root path="remoteJMSObjectFactory.jar"/>
  </resources>

  <dependencies>
    <module name="javax.api"/>
    <module name="javax.jms.api"/>
  </dependencies>
</module>

```

3). Add the generic JMS module as a dependency for all deployments as global modules. For example:

EAP\_HOME/standalone/configuration/standalone-full.xml is used as the JBoss EAP 6 configuration file.

In EAP\_HOME/standalone/configuration/standalone-full.xml, under <subsystem xmlns="urn:jboss:domain:ee:1.1">, add:

```
<global-modules>
  <module name="org.jboss.genericjms.provider" slot="main"/>
  <module name="org.jboss.common-core" slot="main"/>
</global-modules>
```

4). Replace the default HornetQ resource adapter with the generic resource adapter.

In EAP\_HOME/standalone/configuration/standalone-full.xml, replace <subsystem xmlns="urn:jboss:domain:ejb3:1.4"> <mdb>, with:

```
<mdb>
  <resource-adapter-ref resource-adapter-name="org.jboss.genericjms"/>
  <bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>
</mdb>
```

5). Add bindings for your JMS topics and queues as remote objects as necessary.

In EAP\_HOME/standalone/configuration/standalone-full.xml, under <subsystem xmlns="urn:jboss:domain:naming:1.4">, add the bindings, replacing PROVIDER\_QUEUE and PROVIDER\_TOPIC as necessary:

```
<bindings>
  <object-factory name="java:/queue/PROVIDER_QUEUE" module="org.jboss.genericjms.provider"
class="RemoteJMSObjectFactory"/>
  <object-factory name="java:/topic/PROVIDER_QUEUE" module="org.jboss.genericjms.provider"
class="RemoteJMSObjectFactory"/>
</bindings>
```

6). In EAP\_HOME/standalone/configuration/standalone-full.xml, add the generic resource adapter configuration to <subsystem xmlns="urn:jboss:domain:resource-adapters:1.1"> .

Replace PROVIDER\_CONNECTION\_FACTORY, PROVIDER\_HOST, and PROVIDER\_PORT with the JMS provider values.

```
<resource-adapters>
  <resource-adapter id="org.jboss.genericjms">
    <module slot="main" id="org.jboss.genericjms"/>
    <connection-definitions>
      <connection-definition class-
```

```

name="org.jboss.resource.adapter.jms.JmsManagedConnectionFactory" jndi-
name="java:/jms/PROVIDER_CONNECTION_FACTORY" pool-
name="PROVIDER_CONNECTION_FACTORY">
  <config-property name="JndiParameters">
    java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialContextFactory;java.naming.provi
der.url=tcp://PROVIDER_HOST:PROVIDER_PORT
  </config-property>
  <config-property name="ConnectionFactory">
    PROVIDER_CONNECTION_FACTORY
  </config-property>
  <security>
    <application/>
  </security>
</connection-definition>
</connection-definitions>
</resource-adapter>
</resource-adapters>

```

7). The generic JMS resource adapter is now configured and ready for use. Start JBOSS EAP 6.2 using following comamnd:

```
standalone.bat -c standalone-full.xml
```

8). Create a message driven bean (MDB).

Use code similar below to use the resource adapter. Replace PROVIDER\_CONNECTION\_FACTORY, PROVIDER\_HOST, and PROVIDER\_PORT with the JMS provider values.

```

=====
@MessageDriven(activationConfig = {
  @ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue"),
  @ActivationConfigProperty(propertyName = "jndiProperties", propertyValue =
"java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialContextFactory;java.naming.provider.
url=tcp://PROVIDER_HOST:PROVIDER_PORT")
  @ActivationConfigProperty(propertyName = "destination", propertyValue = "PROVIDER_QUEUE"),
  @ActivationConfigProperty(propertyName = "connectionFactory", propertyValue =
"PROVIDER_CONNECTION_FACTORY")
})

@ResourceAdapter("generic-jms-ra.rar")
public class SampleMdb implements MessageListener {
  @Override

```

```
public void onMessage(Message message) {  
  
}  
  
}  
=====
```

If the EMS server's authorization is turned on you will need to set following: @ActivationConfigProperty

For example:

a). username and password for the JNDI lookup:

```
@ActivationConfigProperty(propertyName = "jndiProperties", propertyValue =  
"java.naming.factory.initial=com.tibco.tibjms.naming.TibjmsInitialContextFactory;java.naming.provider.  
url=tcp://PROVIDER_HOST:PROVIDER_PORT;java.naming.security.principal=PROVIDER_JBOSS_JNDI_LO  
OKUP_USER;java.naming.security.credentials=PASSWORD"),
```

b). username and password for creating the EMS connection:

```
@ActivationConfigProperty(propertyName = "user", propertyValue = "PROVIDER_MDBUSER"),  
@ActivationConfigProperty(propertyName = "password", propertyValue = "PASSWORD"),
```