http://hursleyonwmq.wordpress.com/2007/05/29/simplest-sample-applications-using-websphere-mq-jms/

Sample applications using WebSphere MQ JMS

Author: Saket Rungta, wastedmonkeys.com

Simple Point-to-point application using WebSphere MQ JMS

```
package my.samples;

import javax.jms.JMSException;
import javax.jms.Session;

import com.ibm.jms.JMSMessage;
import com.ibm.jms.JMSTextMessage;
import com.ibm.mq.jms.JMSC;
import com.ibm.mq.jms.MQQueue;
import com.ibm.mq.jms.MQQueueConnection;
import com.ibm.mq.jms.MQQueueConnectionFactory;
import com.ibm.mq.jms.MQQueueReceiver;
import com.ibm.mq.jms.MQQueueSender;
import com.ibm.mq.jms.MQQueueSession;

/**
 * SimplePTP: A minimal and simple testcase for Point-to-point messaging (1.02 style).
 *
 * Assumes that the queue is empty before being run.
 *
 * Does not make use of JNDI for ConnectionFactory and/or Destination definitions.
 *
 * @author saket
 */
public class SimplePTP {
  /**
   * Main method
   *
   * @param args
   */
  public static void main(String[] args) {
    try {
      MQQueueConnectionFactory cf = new MQQueueConnectionFactory();

      // Config
      cf.setHostName("localhost");
      cf.setPort(1414);
      cf.setTransportType(JMSC.MQJMS_TP_CLIENT_MQ_TCPIP);
      cf.setQueueManager("QM_thinkpad");
```

```
      cf.setChannel("SYSTEM.DEF.SVRCONN");

      MQQueueConnection connection = (MQQueueConnection) cf.createQueueConnection();
      MQQueueSession session = (MQQueueSession) connection.createQueueSession(false, Session.AUTO_ACKNOWLEDGE);
      MQQueue queue = (MQQueue) session.createQueue("queue:///Q1");
      MQQueueSender sender =  (MQQueueSender) session.createSender(queue);
      MQQueueReceiver receiver = (MQQueueReceiver) session.createReceiver(queue);

      long uniqueNumber = System.currentTimeMillis() % 1000;
      JMSTextMessage message = (JMSTextMessage) session.createTextMessage("SimplePTP "+ uniqueNumber);

      // Start the connection
      connection.start();

      sender.send(message);
      System.out.println("Sent message:\\n" + message);

      JMSMessage receivedMessage = (JMSMessage) receiver.receive(10000);
      System.out.println("\\nReceived message:\\n" + receivedMessage);

      sender.close();
      receiver.close();
      session.close();
      connection.close();

      System.out.println("\\nSUCCESS\\n");
    }
    catch (JMSException jmsex) {
      System.out.println(jmsex);
      System.out.println("\\nFAILURE\\n");
    }
    catch (Exception ex) {
      System.out.println(ex);
      System.out.println("\\nFAILURE\\n");
    }
  }
}
```

Simple Publish/Subscribe application using WebSphere MQ JMS

```
package my.samples;

import javax.jms.JMSException;
import javax.jms.Session;

import com.ibm.jms.JMSMessage;
import com.ibm.jms.JMSTextMessage;
import com.ibm.mq.jms.JMSC;
import com.ibm.mq.jms.MQTopic;
import com.ibm.mq.jms.MQTopicConnection;
import com.ibm.mq.jms.MQTopicConnectionFactory;
import com.ibm.mq.jms.MQTopicPublisher;
import com.ibm.mq.jms.MQTopicSession;
import com.ibm.mq.jms.MQTopicSubscriber;

/**
 * SimplePubSub: A minimal and simple testcase for Publish/Subscribe (1.02 style).
 *
 * Topics are dynamically created on the queue manager and need not be pre-defined.
 *
 * (The Broker must be enabled on the queue manager and the JMS Publish/Subscribe
 * queues must be defined manually.)
 *
 * Note: These samples are for WMQ Base Broker and incomplete for WebSphere Message Broker)
 *
 * Does not make use of JNDI for ConnectionFactory and/or Destination definitions.
 *
 * @author saket
 */
public class SimplePubSub {
  /**
   * Main method.
   *
   * @param args
   */
  public static void main(String[] args) {
    try {
      MQTopicConnectionFactory cf = new MQTopicConnectionFactory();
```

```
      // Config
      cf.setHostName("localhost");
      cf.setPort(1414);
      cf.setTransportType(JMSC.MQJMS_TP_CLIENT_MQ_TCPIP);
      cf.setQueueManager("QM_thinkpad");
      cf.setChannel("SYSTEM.DEF.SVRCONN");

      MQTopicConnection connection = (MQTopicConnection) cf.createTopicConnection();
      MQTopicSession session = (MQTopicSession) connection.createTopicSession(false, Session.AUTO_ACKNOWLEDGE);
      MQTopic topic = (MQTopic) session.createTopic("topic://foo");
      MQTopicPublisher publisher =  (MQTopicPublisher) session.createPublisher(topic);
      MQTopicSubscriber subscriber = (MQTopicSubscriber) session.createSubscriber(topic);

      long uniqueNumber = System.currentTimeMillis() % 1000;
      JMSTextMessage message = (JMSTextMessage) session.createTextMessage("SimplePubSub "+ uniqueNumber);

      // Start the connection
      connection.start();

      publisher.publish(message);
      System.out.println("Sent message:\\n" + message);

      JMSMessage receivedMessage = (JMSMessage) subscriber.receive(10000);
      System.out.println("\\nReceived message:\\n" + receivedMessage);

      publisher.close();
      subscriber.close();
      session.close();
      connection.close();

      System.out.println("\\nSUCCESS\\n");
    }
    catch (JMSException jmsex) {
      System.out.println(jmsex);
      System.out.println("\\nFAILURE\\n");
    }
    catch (Exception ex) {
      System.out.println(ex);
      System.out.println("\\nFAILURE\\n");
    }
  }
}
```