

Message Queuing using SMQ

In this practice, two applications will be built. One application translates xCBL PO messages to RosettaNET PO messages. The application will listen to a message queue (SMQ), process the message and store the result in another queue. The transformation to be done is given to you as a XSLT file. Another application will pick files (xCBL PO documents) from a folder and send these to the first application for processing.

The tutorial covers creating the service (on GlassFish) and the client (console application) applications.

This tutorial is inspired by the scenario in an article written by David Chappel: <http://xml.sys-con.com/node/40143>.

Additional Requirements

To complete this tutorial, you need a GlassFish server installed.

You also need to download [JAXP](#).

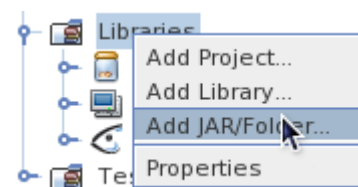
Creating a message queue

You can create a message queue in GlassFish Administration Console or from NetBeans if you have GlassFish installed with NetBeans on your own computer. To create a new queue from the Administration Console, follow these steps:

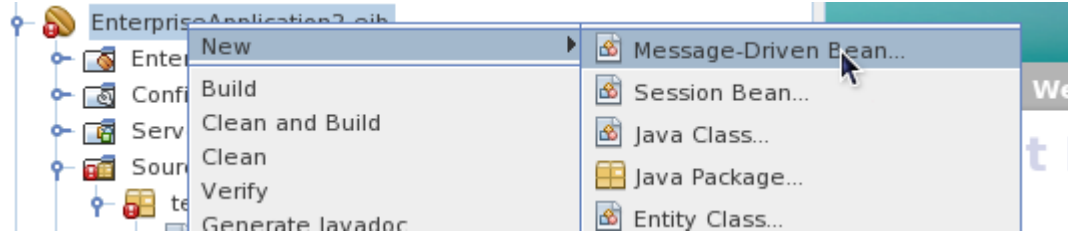
- Open the Admin Console.
- Log in as administrator.
- Open *Resources/JMS Resources/Connection Factories*.
- Click New... to create a new connection factory:
 - JNDI Name: *jms/usernameConnectionFactory*, where username is your username.
 - Type: *javax.jms.ConnectionFactory*
- Open Destination Resources and add a new destination:
 - JNDI Name: *jms/usernameCblQueue*, where username is your username.
 - Physical Destination Name: *usernameCblQueue*, where username is your username.
 - Type: *javax.JMS.Queue*
- Open Destination Resources and add a new destination:
 - JNDI Name: *jms/usernameRosettaQueue*, where username is your username.
 - Physical Destination Name: *usernameRosettaQueue*, where username is your username.
 - Type: *javax.JMS.Queue*

Create a SMQ service using NetBeans

- In NetBeans create a new Enterprise Application Project:
 - Let it create an ejb module as well.
- Add JAXP API library to the service bean.
- Copy XSL file to the project.
- Add new Message-Driven Bean:



- Set the Project Destination to **usernameCblQueue**, where username is your username.



- Right click on the source window and choose Enterprise Resources > Send JMS Message.
 - From Server Destinations choose jms/usernameRosettaQueue.
 - Implement createJMSMessageFor... method by adding code that returns TextMessage with messageData as its contents.
- Edit the onMessage method:
 - Verify that the message is a TextMessage.
 - Create new Source (StreamSource from xsl file).
 - Create a new Transformer using TransformerFactory (in javax.xml.transform namespace, give the transformation source as the xslt file source).
 - Create new Source (StreamSource from StringStream from message text).
 - Create new empty DOMResult object.
 - Transform the file.
 - Send the transformation result (you need to get node from result, get nodes document node and convert it into string) to the new queue.
- Build and deploy your application (remember to add your username to the service name to avoid clashes).

Create SMQ client using NetBeans

- Create a new Enterprise Application Client project.
- Add code to send a new JMS Message to jms/usernameCblQueue.
- Implement the createMessage... method.
- Add code to main method to send the message.