

Using Queues with Delivery Failure

Some days ago I realized that when a message is created through the Admin Console the Queue parameters set to use an Error Destination do not override the message's parameters, it made me lose time because I thought that my application had a problem. However, the problem was caused because the "Redelivery Limit" set on the message by the Admin Console was no overwritten by the "Redelivery Limit" set in the Queue.

When a JMS Queue is configured to send messages with problems to an "Error Destination", it is important to configure a property called "Redelivery Limit" to avoid infinite loops that happen when the message is resend without limits. The following screens show the configuration used.

Configuration used

1. This post was created using Oracle Weblogic 12.1.3
2. A JMS Server is created, this server is shown here.

JMS Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

New Delete Showing 1 to 1 of 1 Previous | Next

<input type="checkbox"/>	Name	Persistent Store	Target	Current Target	Health
<input type="checkbox"/>	JMSServerTest		WLS_01	WLS_01	OK

New Delete Showing 1 to 1 of 1 Previous | Next

As can be seen in the previous figure, the JMS Sever is targeted on the managed server WLS_01.

3. A JMS Module called "JMSModuleTest" is created

JMS Modules

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

New Delete Showing 1 to 1 of 1 Previous | Next

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	JMSModuleTest	System

New Delete Showing 1 to 1 of 1 Previous | Next

- A subdeployment is created and targeted to the JMS Server called "JMSServerTest"

<input type="checkbox"/>	Name ^	Resources	Targets
<input type="checkbox"/>	Prueba		JMSServerTest
<input type="checkbox"/>	SubDeployTest		

- A connection factory called "TestNonUDDCF" is created within the module JMSServerTest. In addition, this connection factory is linked to the subdeployment called "Prueba"

<input type="checkbox"/>	Name ^	Type	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	TestNonUDDCF	Connection Factory	jms/TestNonUDDCF	Prueba	JMSServerTest

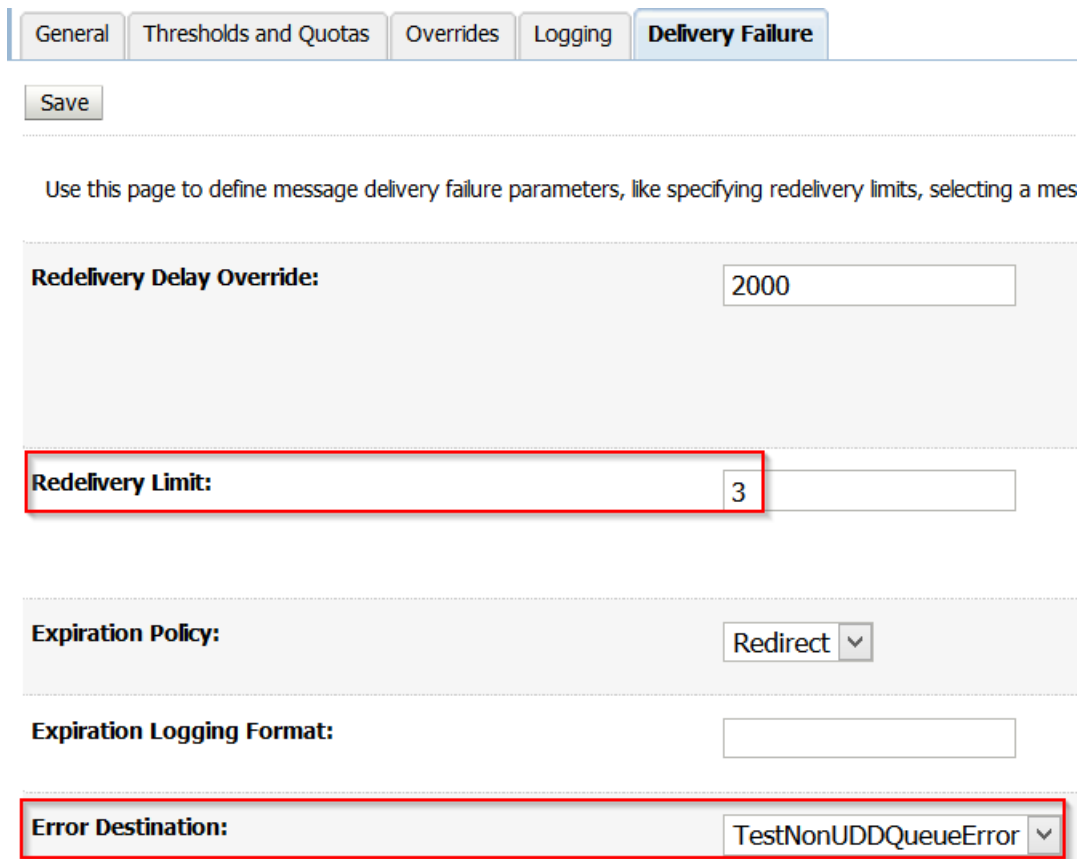
- A queue called "TestNonUDDQueue" to receive the messages is created and linked to the subdeployment called "Prueba".

<input type="checkbox"/>	Name	Type ^	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	TestNonUDDCF	Connection Factory	jms/TestNonUDDCF	Prueba	JMSServerTest
<input type="checkbox"/>	TestNonUDDQueue	Queue	jms/TestNonUDDQueue	Prueba	JMSServerTest

- A queue called "TestNonUDDQueueError" to receive the messages with problems is created and linked to the subdeployment called "Prueba"

<input type="checkbox"/>	Name	Type ^	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	TestNonUDDCF	Connection Factory	jms/TestNonUDDCF	Prueba	JMSServerTest
<input type="checkbox"/>	TestNonUDDQueue	Queue	jms/TestNonUDDQueue	Prueba	JMSServerTest
<input type="checkbox"/>	TestNonUDDQueueError	Queue	jms/TestNonUDDQueueError	Prueba	JMSServerTest

8. It is important to remark that the Queue “TestNonUDDCF” is configured to send messages with problems to the queue “TestNonUDDQueueError”



General Thresholds and Quotas Overrides Logging **Delivery Failure**

Save

Use this page to define message delivery failure parameters, like specifying redelivery limits, selecting a mes

Redelivery Delay Override: 2000

Redelivery Limit: 3

Expiration Policy: Redirect

Expiration Logging Format:

Error Destination: TestNonUDDQueueError

In the previous picture the Redelivery Delay Override is set to 2000 s and the Redelivery Limit is set to 3, which means that a message with problems will be redelivered each 2 seconds during 3 times before being sent to the “Error Destination”. In this case the queue called “TestNonUDDQueueError”.

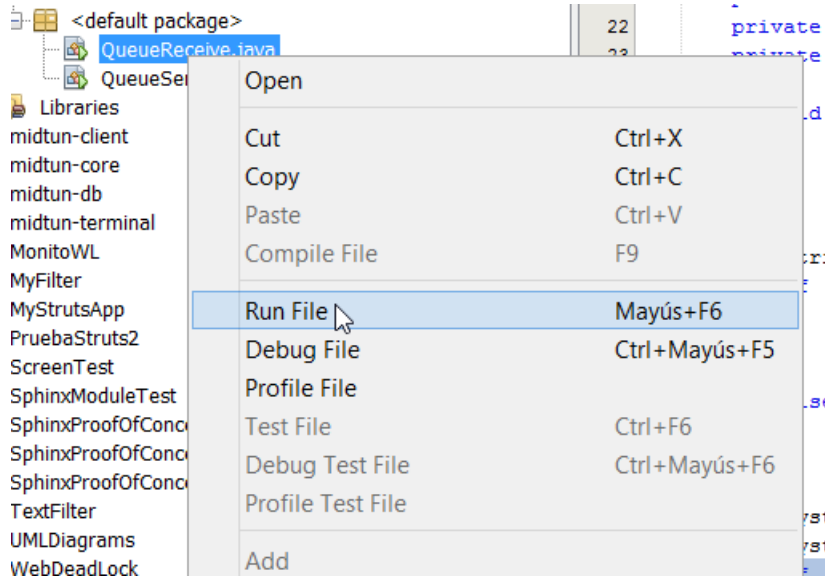
Testing the configuration deployed

In order to test this configuration a Java client in NetBeans is used to read messages from the queue called “TestNonUDDCF”. In case the message is equal to “rollback” the process is cancelled to send the message to the Error Destination.

```
if (msgText.equalsIgnoreCase("rollback"))
{
    // msg.setJMSRedelivered(false);
    qsession.rollback();
}
```

These are the steps followed during the test.

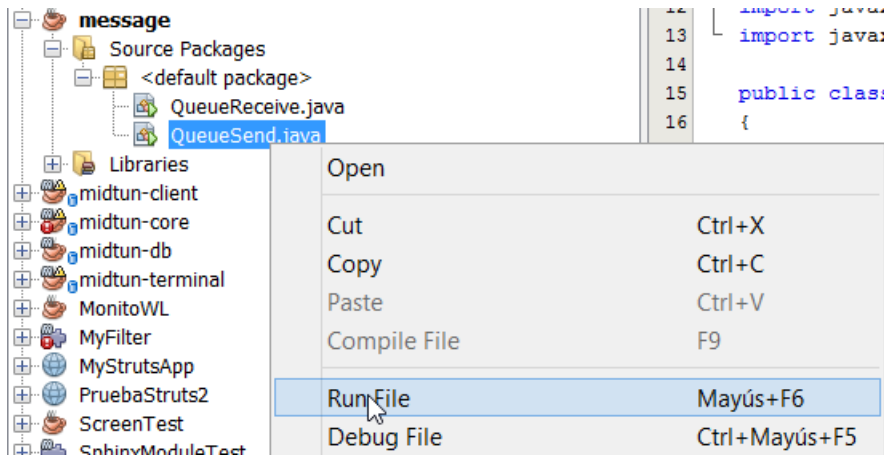
1. Execute the Java class called "QueueReceive"



2. Create some messages using the Admin Console

A screenshot of a web-based 'Produce JMS Message' dialog box. The dialog has a title bar 'Produce JMS Message' and 'OK' and 'Cancel' buttons. Below the title bar, it says 'JMS Message' and 'The following properties will be used to produce a JMS message.' There are several input fields: 'Type:', 'Correlation ID:', 'Expiration:', 'Priority:' (with a dropdown menu showing '4'), 'Delivery Mode:' (with a dropdown menu showing 'Persistent'), 'Delivery Time:' (with a text input field containing '-1'), and 'Redelivery Limit:' (with a text input field containing '-1'). At the bottom, there is a 'Body:' label and a text area containing 'Hello world 1'. The text area is highlighted with a red border. There are 'OK' and 'Cancel' buttons at the bottom of the dialog.

- Now a class called “QueueSend” is used to create a message in the queue using the word “rollback.”



- In this case the property “Redelivery limit” is set according to the configuration message.

Message created using the Java client **Message created using the Admin Console**

Showing 1 to 2 of 2 Previous | Next

	State String	JMS Delivery Mode	Message Size	JMS Expiration	JMS Redelivered	JMS Redelivery Limit
:52 COT 2016	visible	Persistent	8	0	true	-1
:23 COT 2016	visible	Persistent	8	0	false	3

Red arrows point from the labels above to the 'JMS Redelivered' and 'JMS Redelivery Limit' columns in the table.

- This time the message is redelivered three times before being sent to the Error Destination

```

message (run) × message (run) #2 ×
run:
Usage: java QueueReceive WebLogicURL
Use: t3://syspwks024:7003/
JMS Ready To Receive Messages (To quit, send a 'quit' message from QueueSender.class).
Message.getJMSExpiration: 0
Message.getJMSRedelivered: false
nt&lt;Msg_Receiver&gt; rollback
msg.getJMSMessageID ID:<498441.1460749020305.0>
Message.getJMSExpiration: 0
Message.getJMSRedelivered: true
nt&lt;Msg_Receiver&gt; rollback
msg.getJMSMessageID ID:<498441.1460749020305.0>
Message.getJMSExpiration: 0
Message.getJMSRedelivered: true
nt&lt;Msg_Receiver&gt; rollback
msg.getJMSMessageID ID:<498441.1460749020305.0>
Message.getJMSExpiration: 0
Message.getJMSRedelivered: true
nt&lt;Msg_Receiver&gt; rollback
msg.getJMSMessageID ID:<498441.1460749020305.0>

```

The last three lines of the output are enclosed in a red box, showing the message being redelivered three times.

In conclusion, you should be careful when this functionality is tested since it could make you think there is an error with your application. However, when the message is created using a client different from the Admin Console, the behaviour is the expected.