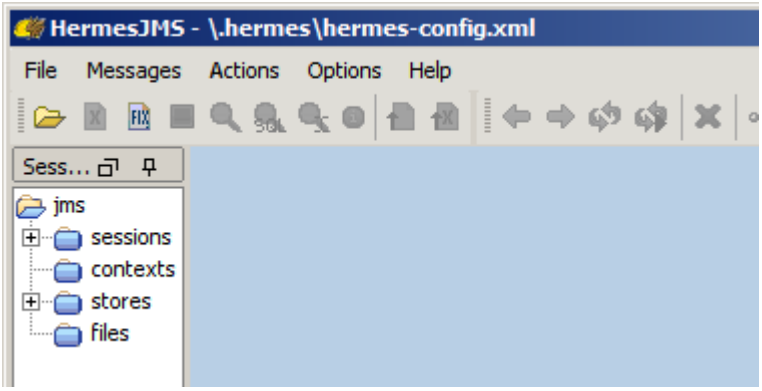
Hermes JMS Configuration for Java MQ

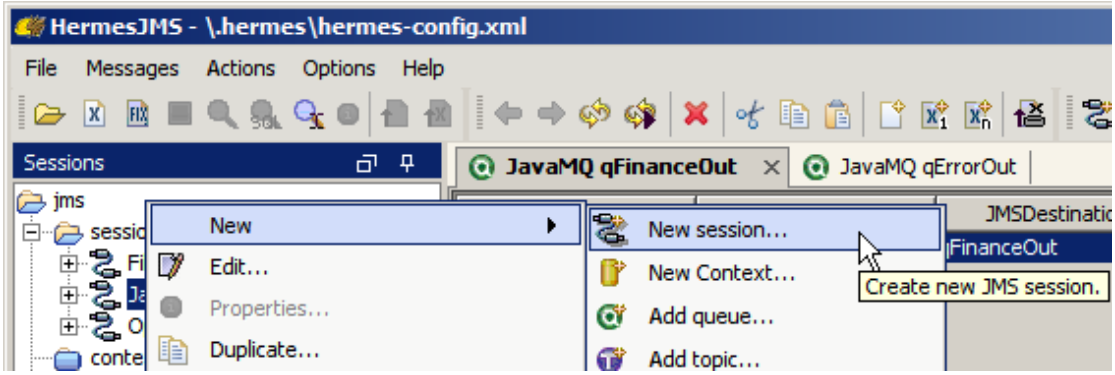
Michael.Czapski@sun.com
November 2008/March 2009
Version 1.1 ☺

Hermes JMS Home Page is at http://www.hermesjms.com/confluence/display/HJMS/Home. There are installation instructions at the site. Follow the installation instructions to get the product installed, then follow the instructions below to configure it to work with Java MQ. By default it will only work if the Java MQ is installed to use the default port 7676. There are additional instructions on how to configure it to communicate with JMQ instances listening on non-default ports.

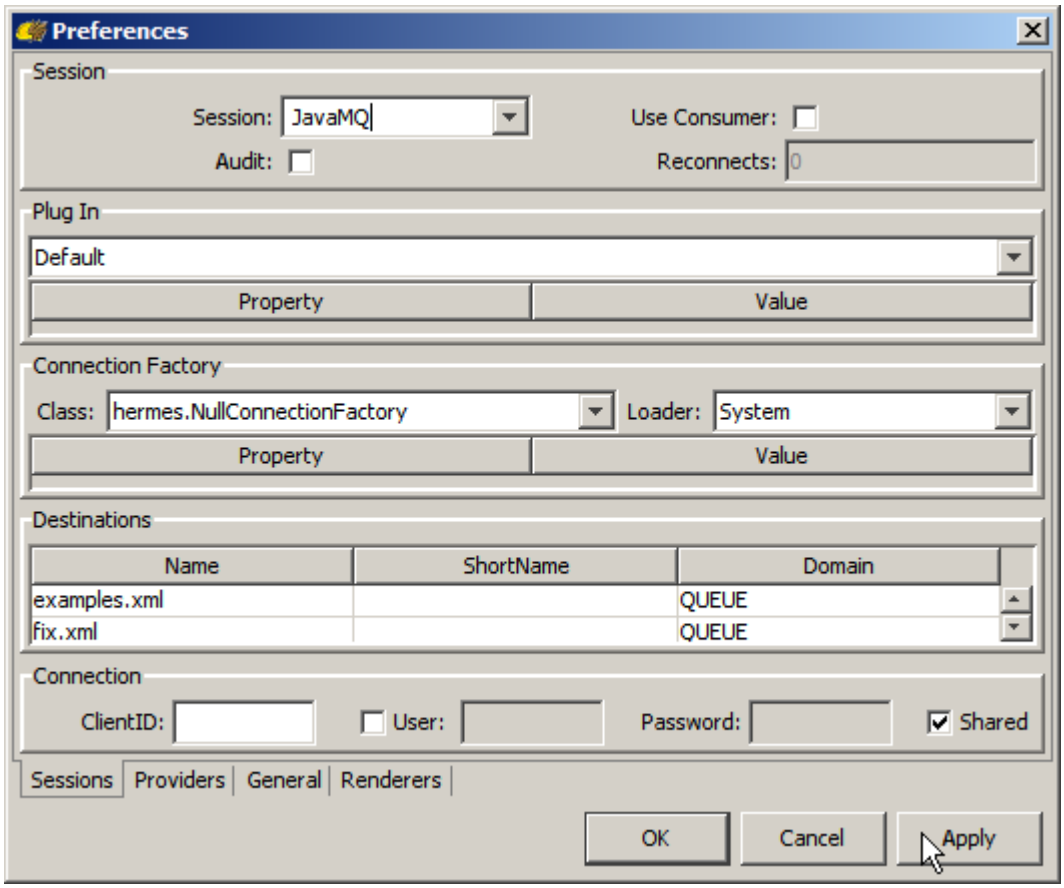
Start HermesJMS. You should see a window similar to the fragment here.



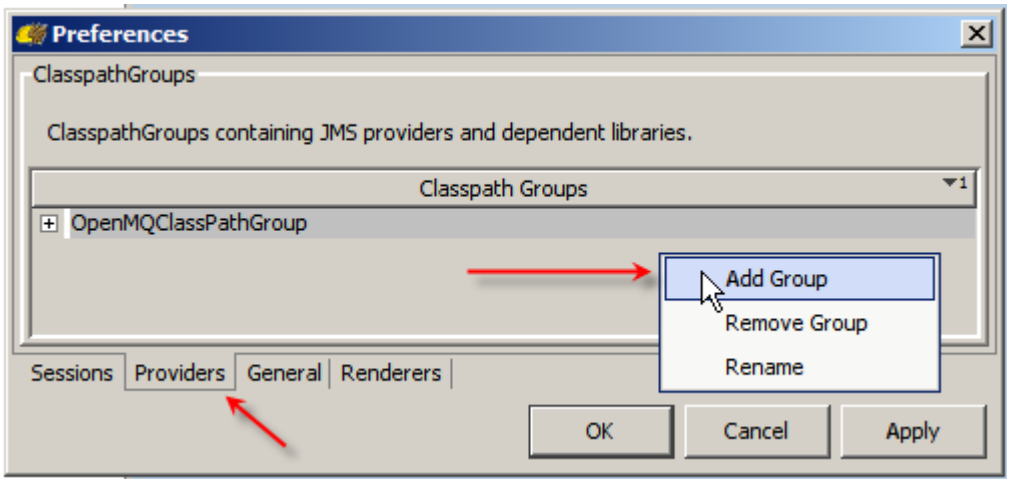
Expand the sessions pane so it is a bit wider then what it is by default and right-click anywhere inside the Sessions pane. Choose New->New session ...



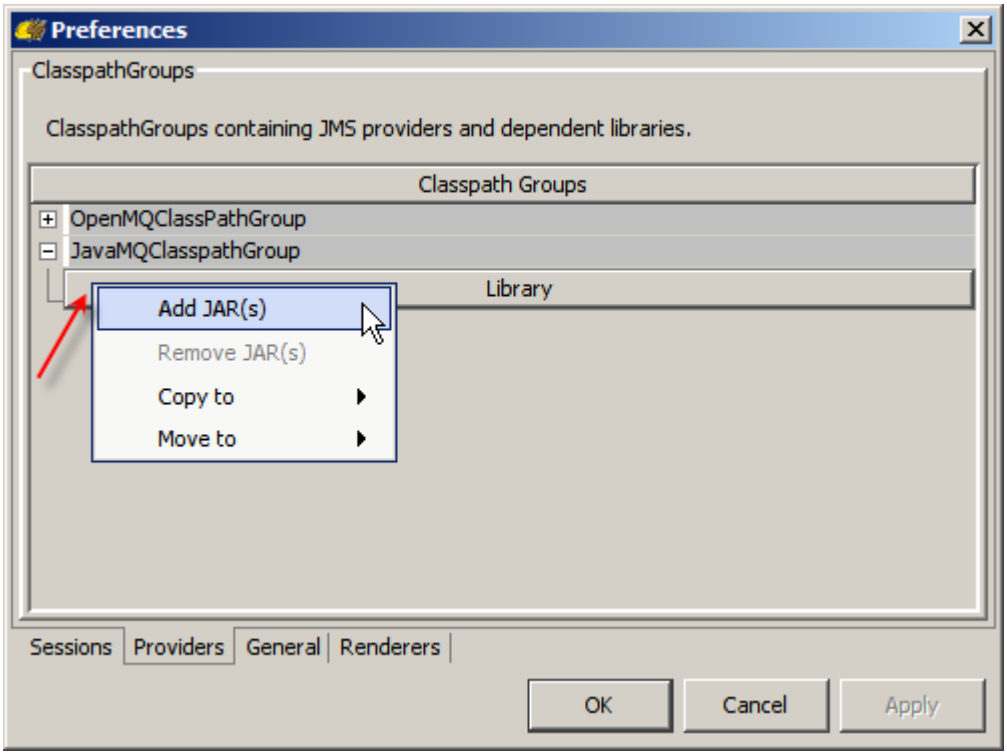
Replace <new> with JavaMQ and click the Apply button



Click on the Providers Tab, right click anywhere inside the Classpath Groups pane and choose Add Group

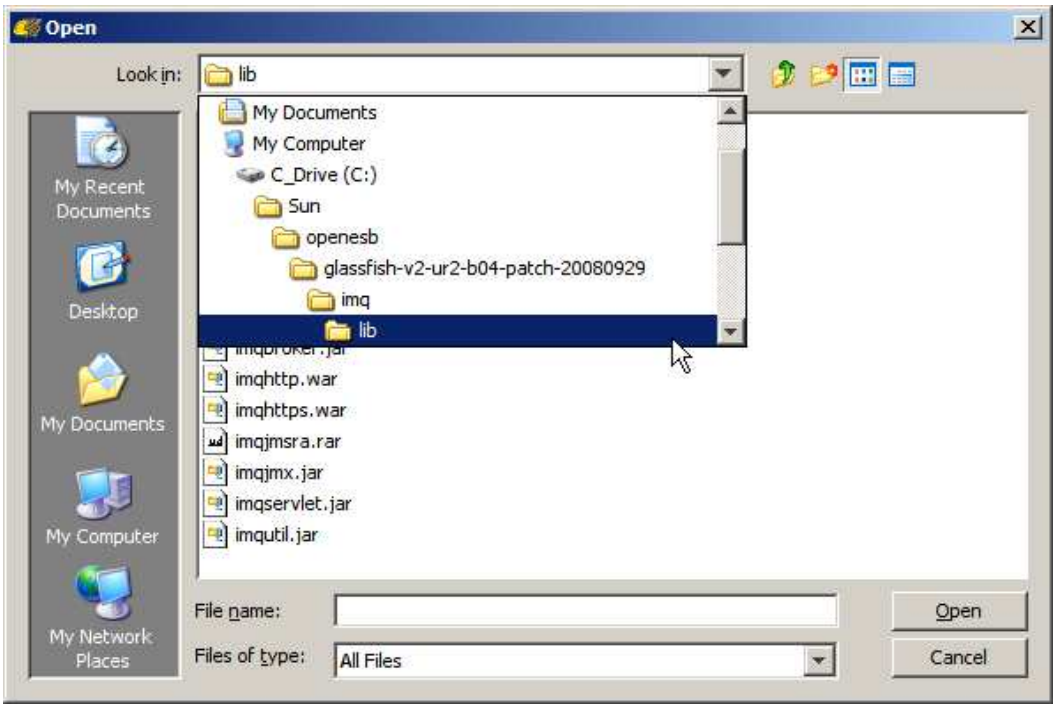


Name the group JavaMQClasspahGroup. Right-click on the long button called Library and choose Add JAR(s).



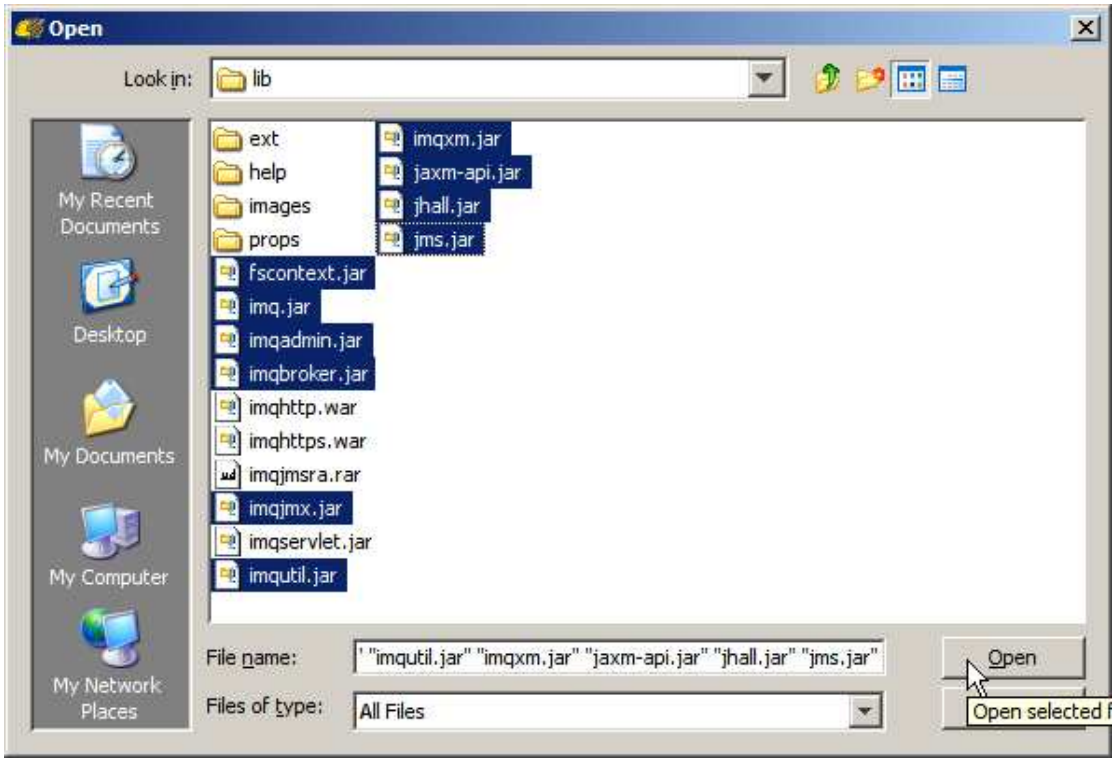
Navigate to your GlassFish application server's imq/lib directory, or your standalone Java MQ installation's lib directory. For OpenESB installed to C:/sun on Windows it will be something like:
C:\Sun\openesb\glassfish-v2-ur2-b04-patch-20080929\imq\lib

This should work whether the imq/lib comes from Java MQ 4.1, OpenESB installation, GlassFishESB installation or Java CAPS 6 installation.

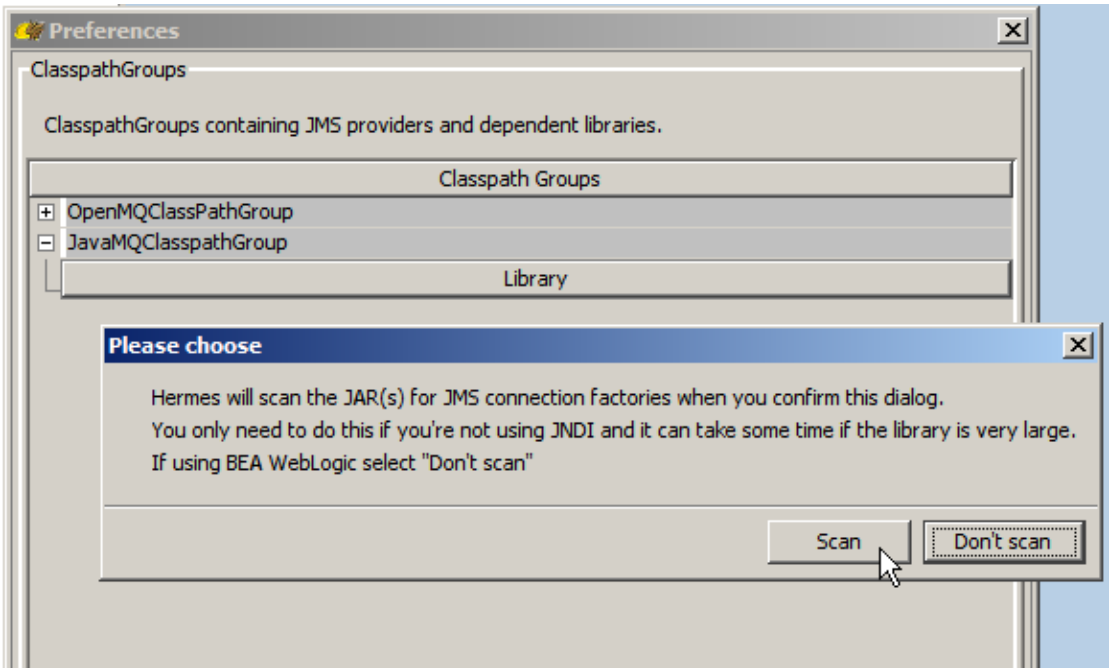


Choose the following JARs. Not all may be required by I did not spend the time figuring out the minimum necessary set.

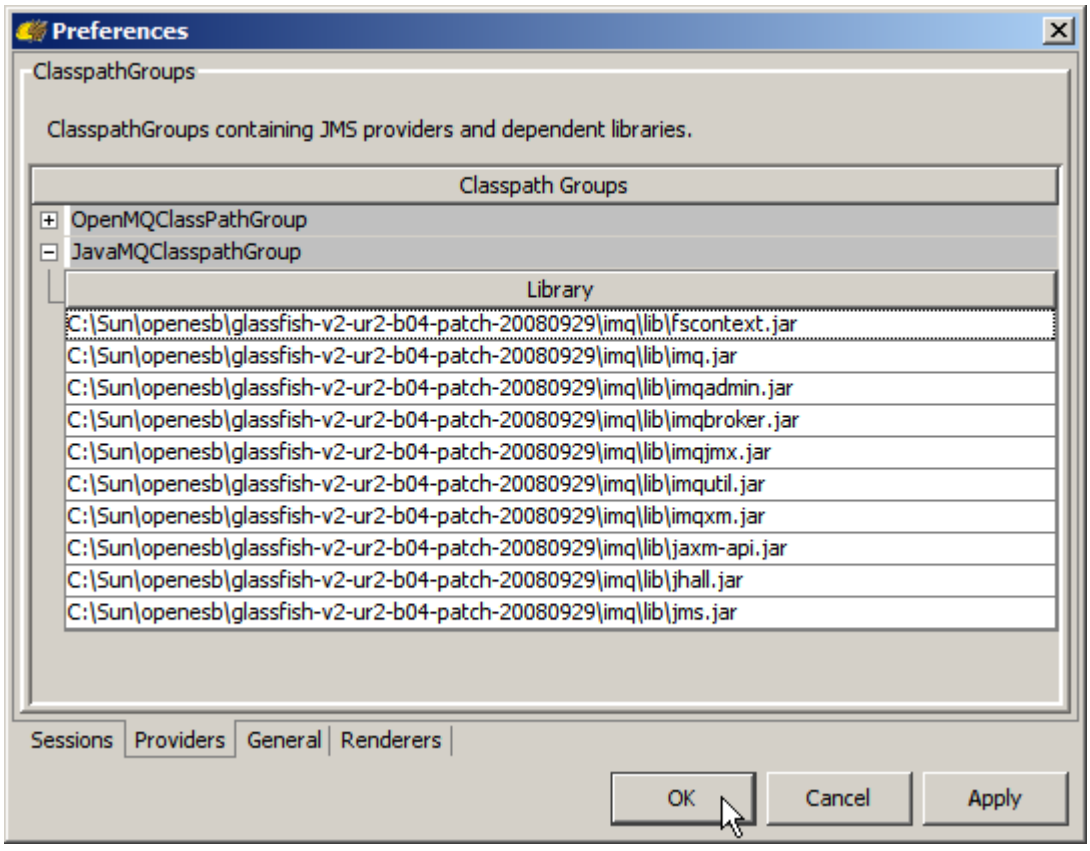
fscontext.jar
imq.jar
imqadmin.jar
imqbroker.jar
imqjmx.jar
imqutil.jar
imqxm.jar
jaxm-api.jar
jhall.jar
jms.jar



Allow Hermes to scan the JARs.

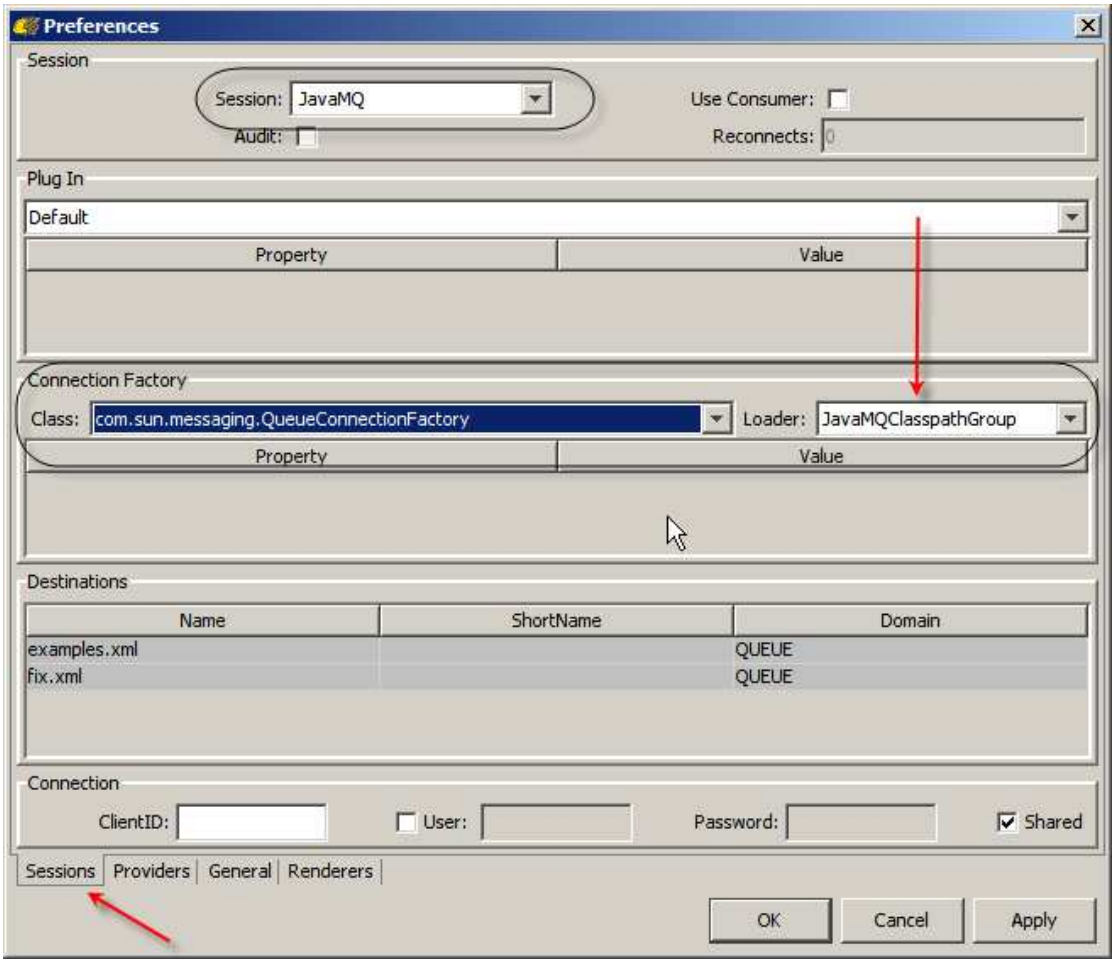


Once the JARs are scanned the Library will list all the JARs. Click the Apply button.

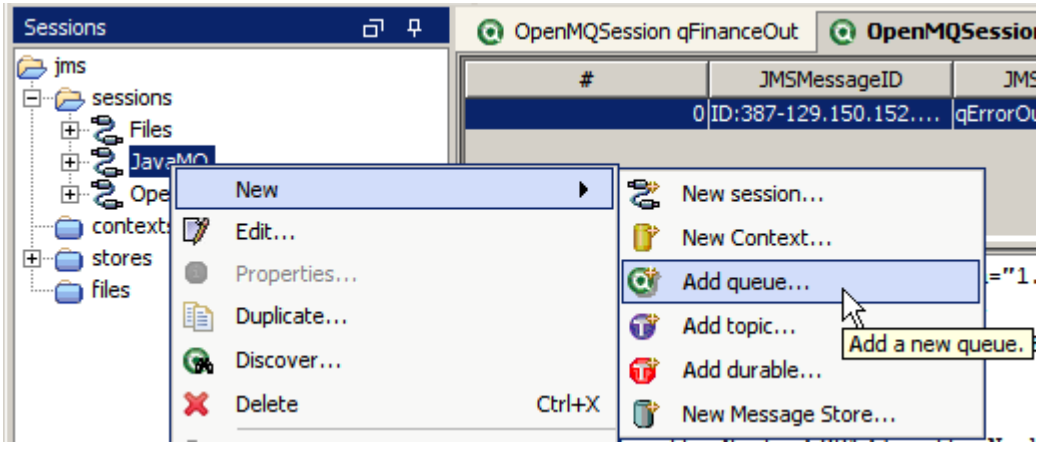


Click the Apply and the OK buttons.
Right-click the JavaMQ session and choose Edit to open configuration dialogue again.

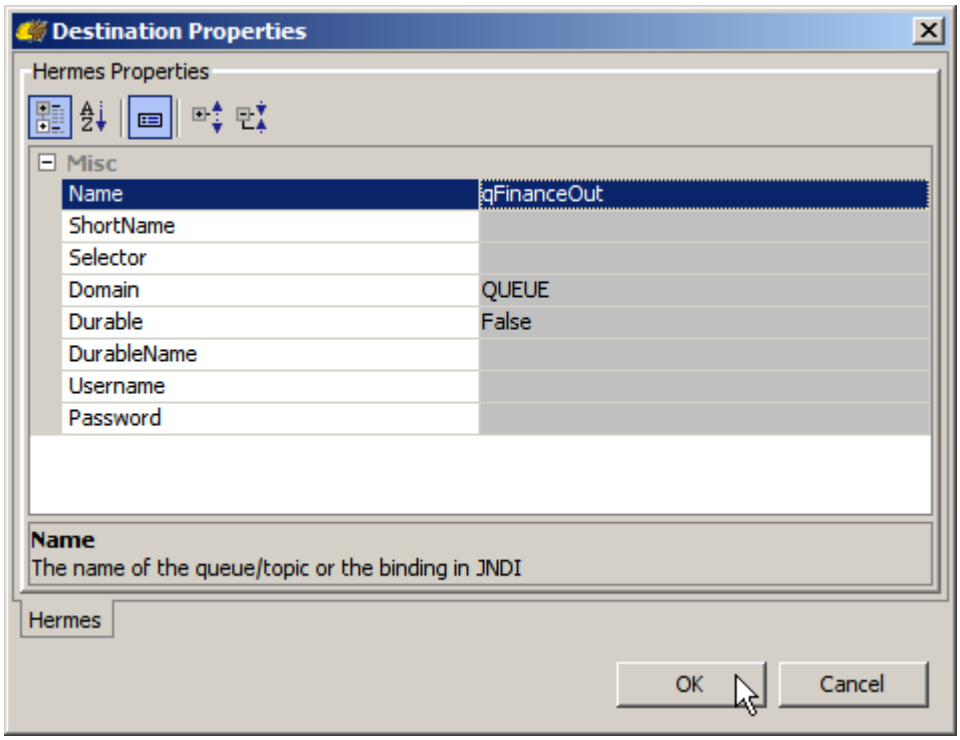
Pull down the Loader menu and choose the JavaMQClasspathGroup. Pull down the Class menu and choose the com.sun.messaging.QueueConnectonFactory class. Click OK.



Right-click JavaMQ session and choose New->Add queue ...



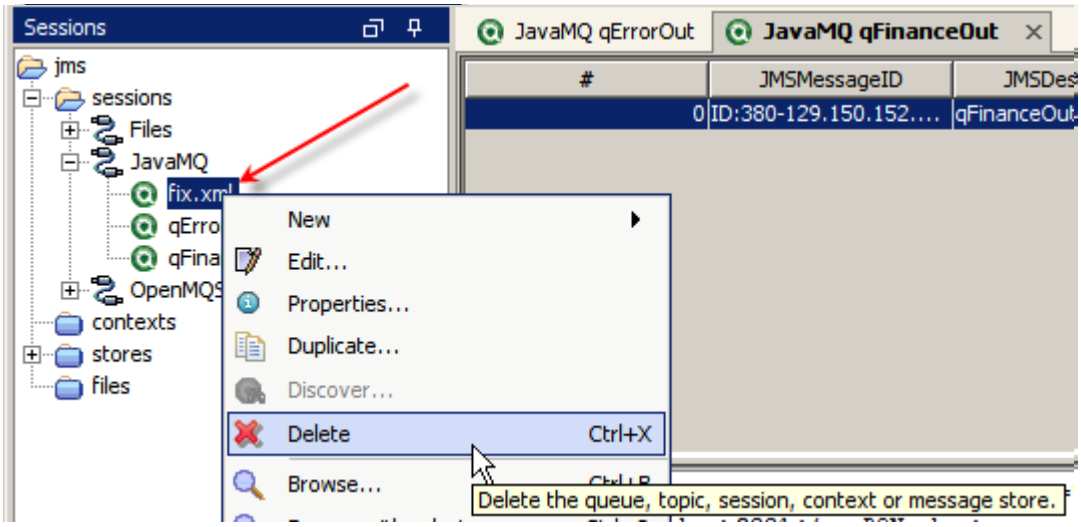
Add queue qFinanceOut, or whatever Queue you need to snoop on.



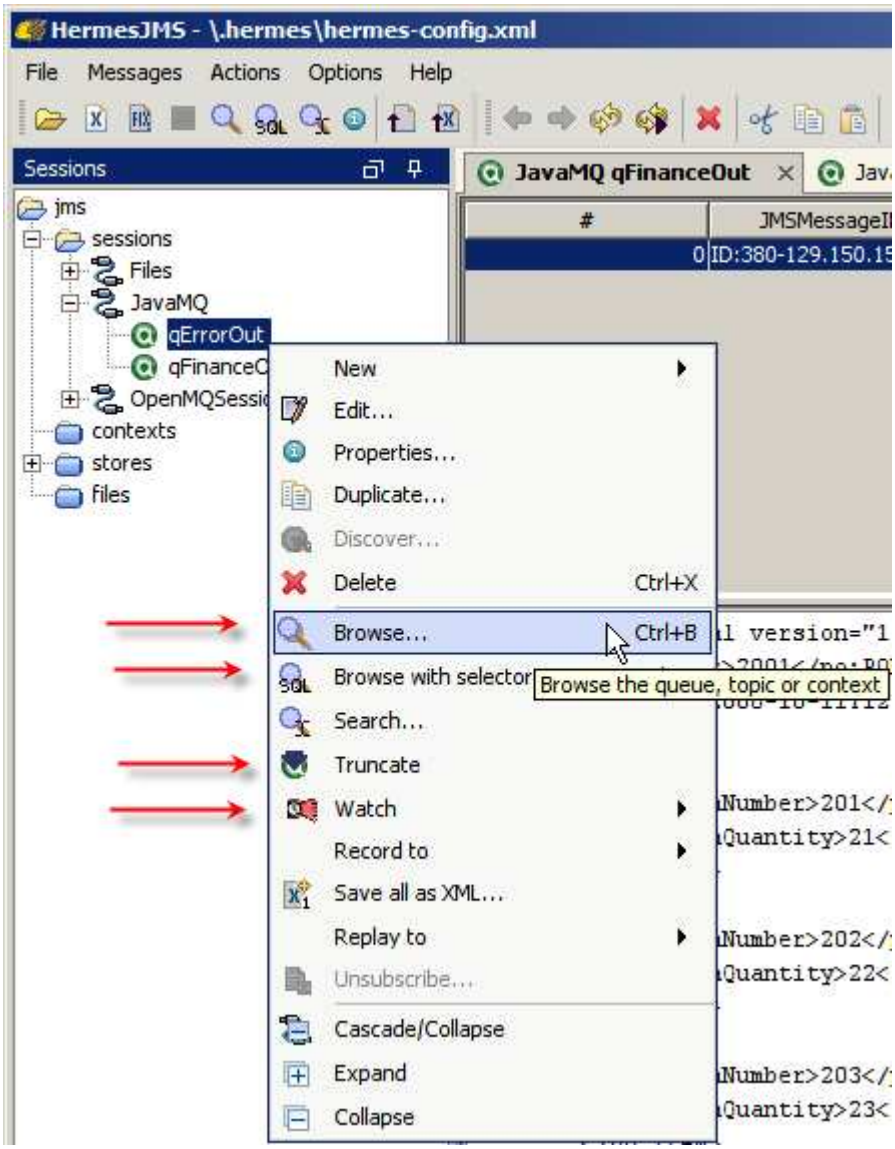
Right click on the JavaMQ session and choose New->Add topic ...

Add topic tMyTopic, or whatever topic you would like to snoop on.

Delete fix.xml and examples.xml under JavaMQ if you like.



Once the JavaMQ session is configured and queues and topics of interest are configured we can interact with the Java MQ broker, view messages in destinations, truncate destinations (delete all messages) and so on. The functionality of Hermes JSM can be used as required.



The material below was not in the original document.

Rajan Jain motivated me to figure out, at least partially, how to configure Hermes JMS to communicate with Java MQ listeners listening on non-default ports.

In the HermesJMS {installation directory}/bin/hermes.bat (on Windows or equivalent on Unix) there is a command that launches Hermes JMS application class. Add a java system property like:

```
-DimqAddressList=mq://localhost:7677,mq://localhost:7676
```

somewhere amongst other system properties there, once you configured Java MQ as discussed above. Hermes JMS will try each server in turn to locate the queue/topic you select to operate on. I have not spend the time thinking how one would make it resolve queues/topics with the same names on different servers.

Example from by script looks like:

```
start "HermesJMS" "%JAVA_HOME%\bin\javaw" -XX:NewSize=256m -Xmx1024m
-Dhermes.home="%HERMES_HOME%" %HERMES_OPTS% -Dlog4j.configuration="fi
le:%HERMES_HOME%\bin\log4j.props" -DimqAddressList=mq://localhost:767
7,mq://localhost:7676 -Dsun.java2d.noddraw=true -Dhermes="%HERMES_CON
FIG%\hermes-config.xml" -Dhermes.libs="%HERMES_LIBS%" hermes.browser.
HermesBrowser
```

Subsequently, Isa Hashim contributed the following information in a comment to my original blog post, which I am reproducing here without permission (sorry Isa, I though it important to let people have this information):

> A number of people have been asking about HermesJMS/Open MQ on the MQ forum:
>
> http://forums.sun.com/forum.jspa?forumID=711
>
> and on users@mq.dev.java.net for a while now - thanX very much for your useful post.
>
> I've just written up and placed a FAQ at:
>
> http://wiki.glassfish.java.net/Wiki.jsp?page=OpenMQHermesJMSQuestions
>
> which contains similar steps to yours plus some steps for configuring HermesJMS via JNDI. I'm currently also looking at creating an OpenMQ plugin for HermesJMS so that HermesJMS among other things can 'discover' the destinations on the MQ broker instead of the user having to know the destinations to browse; the above FAQ page will be updated when that work materializes.
>
> With regards to connecting to a remote broker, I was only succesful by doing it via the JNDI way. I tried to add properties to the connection factory (eg {imqAddressList=mq://localhost:7677}) after selecting it on the Preferences dialog:
>
> http://wiki.glassfish.java.net/attach/OpenMQHermesJMSQuestions/new_session_choose_cf.gif
>
> but it didn't work. I'll keep looking... Let me know if you manage to get it to work other than using '-DimqAddressList=...' on the command line - I would like to get this to work so that I can connect to multiple remote brokers at the same time and browse/copy msgs to/from them.
>
> regards,
> -isa