OpenMQ

Starting the broker

/opt/OpenMessageQueue/mq/bin/imqbrokerd &

/opt/OpenMessageQueue/mq/bin/imqadmin

Add your broker on localhost:7676. The user is admin and the password is admin. Then connect to it.

A destination called mq.sys.dmq will appear in the Destinations panel

Configure JNDI

Connection Factory

You can either use LDAP or a local directory to store the java references.

Create a config file for imqobjmgr

```
version=2.0
cmdtype=add
obj.type=cf
obj.lookupName=MyConnectionFactory
obj.attrs.imqAddressList=mq://host:7676/jms
objstore.attrs.java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory
objstore.attrs.java.naming.provider.url=file:///tmp/jndi
```

The imqAddressList attribute is needed if the JMS client will be running on another host.

Register the connection factory object

mkdir /tmp/jndi
/opt/OpenMessageQueue/mq/bin/imqobjmgr -i add_cf.props

Destination

Do the same for a queue object

```
version=2.0
cmdtype=add
obj.type=q
obj.lookupName=Q1
obj.attrs.imqDestinationName=QueueOne
objstore.attrs.java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory
objstore.attrs.java.naming.provider.url=file:///tmp
```

/opt/OpenMessageQueue/mq/bin/imqobjmgr -i add_q.props

If you examine file /tmp/jndi/.bindings you can see the java naming reference objects and the String RefAddr information.

A JMS client needs to use the following naming properties to fetch the connection factory and queue objects

java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory
java.naming.provider.url=file:///tmp/jndi

If not running the JMS client on the same host as the broker transfer the /tmp/jndi/.bindings file from the broker host to the local host. The JARs needed are in directiry /opt/OpenMessageQueue/mq/lib

java -classpath .:imq.jar:fscontext.jar:jms.jar JMSsend a.xml MyConnectionFactory Q1

```
Start
Getting Initial Context
lookup JNDI=MyConnectionFactory
class = com.sun.messaging.ConnectionFactory
Getting Queue JNDI=Q1
class = com.sun.messaging.Queue
Getting Connection for Queue
staring the connection
creating session
creating messageProducer
creating TextMessage
sending Message
```

HermesJMS

Create a provider in the preferences panel called OpenMQ and add the following classes

imq.jar
fscontext.jar

Create a new context called OpenMQ

Set
loader to be OpenMQ
initalContextFactory to be com.sun.jndi.fscontext.RefFSContextFactory
providerURL to be file:///tmp/jndi

Browse the context. The Queue and Connection Factory objects will appear (along with the .bindings file itself).

Drag the Connection Factory object onto the sessions folder in the navigation panel and give it a session name like OpenMQ. Click the discover queues button or drag the queue object from the context onto the newly created session.

You can now send messages to the queue.