Compile the source

Source is available from the HOL page at this link
set CLASSPATH to {glassfish-home-dir}/lib/javaee.jar:{mq-home-dir}/lib/imq.jar:{mq-home-dir}/lib/jms.jar:.
(CLASSPATH considerations for windows users will need to follow the directions given in class, Verify your system path points to the right Java runtime)
Compile the MDB

```
javac GenericMDB.java
jar cvf ejb-jar-ic.jar GenericMDB.class
jar cvf simplemdb.ear ejb-jar-ic.jar
```

Compile JMS Sender/Receiver

```
javac Sender.java Receiver.java
```

MDB Single Instance Lab

#	Command	Comments
1	`asadmin start-domain`	Start the domain
2	`asadmin create-local-instance instance1`	Create the GlassFish instance
3	`asadmin create-jms-resource --target instance1 --restype javax.jms.Queue --property imqDestinationName=inboundQueue inboundQueue`	Create the JMS inboundQueue destination
4	`asadmin create-jms-resource --target instance1 --restype javax.jms.Queue --property imqDestinationName=outboundQueue outboundQueue`	Create the JMS outboundQueue destination
5	`asadmin create-jms-resource --target instance1 --restype javax.jms.QueueConnectionFactory outboundQueueFactory`	Create the queue connection factory
6	`asadmin deploy --target instance1 /tmp/simplemdb.ear`	Deploy the mdb EAR that you built (directory may need to be changed)
7	`java -DimqBrokerHostPort=27676 Receiver`	Start the Java receiver application – it will use port 27676 and receive messages from outboundQueue (run this from the directory you

#	Command	Comments
		originally compiled your java application in)
	If using a terminal, best to create another terminal window	
8	`java -DimqBrokerHostPort=27676 Sender`	Start the message sender application. It sends messages to inboundQueue. If all goes well, you should immediately see 10 messages sent and the receiver should quickly receive 10 messages. (run this from the directory you originally compiled your java application in)

Cleaning up

#	Command	Comments
1	`asadmin undeploy --target instance1 simplemdb`	Undeploy the MDB
2	`asadmin delete-jms-resource --target instance1 inboundQueue`	Delete the inboundQueue
3	`asadmin delete-jms-resource --target instance1 outboundQueue`	Delete the outbound queue
4	`asadmin delete-jms-resource --target instance1 outboundQueueFactory`	Delete the queue connection factory
5	`asadmin delete-local-instance instance1`	Delete the instance
6	`asadmin stop-domain`	Stop the domain

Cluster MDB Lab

#	Command	Comments
1	`asadmin start-domain`	Start the domain
	`asadmin create-cluster cluster1`	Create a cluster, named cluster1
	`asadmin create-local-instance --cluster cluster1 --systemproperties HTTP_LISTENER_PORT=1111:HTTP_SSL_LISTENER_PORT=1112:IIOP_SSL_LISTENER_PORT=1113:IIOP_LISTENER_PORT=1114:JMX_SYSTEM_CONNECTOR_PORT=1115:IIOP_SSL_MUTUALAUTH_PORT=1116:JMS_PROVIDER_PORT=1117:ASADMIN_LISTENER_PORT=1118 instance1`	Create instance1, in cluster1
	`asadmin create-local-instance --cluster cluster1 --systemproperties HTTP_LISTENER_PORT=2221:HTTP_SSL_LISTENER_PORT=2222:IIOP_SSL_LISTENER_PORT=2223:IIOP_LISTENER_PORT=2224:JMX_SYSTEM_CONNECTOR_PORT=2225:IIOP_SSL_MUTUALAUTH_PORT=2226:JMS_PROVIDER_PORT=2227:ASADMIN_LISTENER_PORT=2228 instance2`	Instance2
	`asadmin create-local-instance --cluster cluster1 --systemproperties HTTP_LISTENER_PORT=3331:HTTP_SSL_LISTENER_PORT=3332:IIOP_SSL_LISTENER_PORT=3333:IIOP_LISTENER_PORT=3334:JMX_SYSTEM_CONNECTOR_PORT=3335:IIOP_SSL_MUTUALAUTH_PORT=3336:JMS_PROVIDER_PORT=3337:ASADMIN_LISTENER_PORT=3338 instance3`	Instance 3
	`asadmin list-instances` `asadmin list-clusters`	To view the current status
	`asadmin start-cluster cluster1`	Start cluster1, rerun list-instances and list-clusters to confirm the cluster and all three instances are running
	`asadmin create-jms-resource --target cluster1 --restype javax.jms.QueueConnectionFactory outboundQueueFactory`	Create the queue connection factory
	`asadmin create-jms-resource --target cluster1 --restype javax.jms.Queue --property imqDestinationName=outboundQueue outboundQueue`	Create the outbound queue destination
	`asadmin create-jms-resource --target cluster1 --restype javax.jms.Queue --property imqDestinationName=inboundQueue inboundQueue`	Create the inbound queue destination
	`asadmin list-jms-resources cluster1`	Verify the JMS resources were created
	`asadmin deploy --target cluster1 simplemdb.ear`	Deploy the same MDB ear that you used in the previous lab

#	Command	Comments
	`imqcmd list dst -b :1117`	Look at the destination (note: you may need to change to the MQ bin directory ${GlassFishInstall/mq/bin)
	`java -DimqBrokerHostPort=1117 Sender`	Send the JMS messages – this port will connect to instance1, defined above. If this succeeds it will produce 10 messages onto inboundQueue destination. (run this from the directory you originally compiled your java application in)
	`imqcmd list dst -b :1117`	This will display the the statistics of the destination in instance1. In most cases, the inboundQueue will be empty, as the MDB will have already read the messages and moved them to the outboundQueue destination.
	`imqcmd list dst -b :2227` `imqcmd list dst -b :3337`	This will show you the messages at the other cluster instance destinations
	`java  -DimqBrokerHostPort=1117 Receiver`	Read all the messages. If this succeeds, it will read 10 messages, print them, and exit. You can use the imqcmd list commands from the previous steps to verify that all the messages have been read.

Clean up

#	Command	Comments
1	`asadmin undeploy --target cluster1 simplemdb`	Undeploy the MDB application
	`asadmin delete-jms-resource --target cluster1 inboundQueue`	Delete the inboundQueue destination from the cluster
	`asadmin delete-jms-resource --target cluster1 outboundQueue`	Delete the outboundQueue destination from the cluster
	`asadmin delete-jms-resource --target cluster1 outboundQueueFactory`	Delete the queue connection factory from the cluster
	`asadmin stop-cluster cluster1`	Stop the cluster
	`asadmin list-instances`	Verify that the instances and cluster are stopped

#	Command	Comments
	`asadmin list-clusters`	

	`asadmin delete-local-instance instance3`
	`asadmin delete-local-instance instance2`
	`asadmin delete-local-instance instance1`
		Delete the instances from the cluster

	`asadmin delete-cluster cluster1`	Delete the cluster
	`asadmin stop-domain`	Stop the domain

Alternatives:
Use the GUI to create the cluster and run the java applications from a command shell (you can just use the port defaults, if you do, be sure to check the port assignments and set them correctly when you run the java applications (typically, they will be 27676, 27677, and 27678)
Use Netbeans to compile and run the java bits (simplemdb, Sender and Receiver)
Use a script to create the cluster after you've compiled the Java applications. We provide a script in the source bundle. You will need to edit it, to point to the glassfish home directory)