1. What you’ll need to setup to demonstrate JMS (SOAP/JMS) service virtualization
   * The SV project for demo execution.
   * Modified jbossmq-destinations-service.xml located in \DevTest Root\DemoServer\lisa-demo-server\jboss\server\default\deploy\jms (call the queue names anything you like!)
     + Append the file with the following at the end:

<!-- Added to enablement challenges -->

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=KARLIN.REQUEST.QUEUE">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=KARLIN.RESPONSE.QUEUE">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=KARLIN.REQUEST.QUEUE.PROXY">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=KARLIN.RESPONSE.QUEUE.PROXY">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

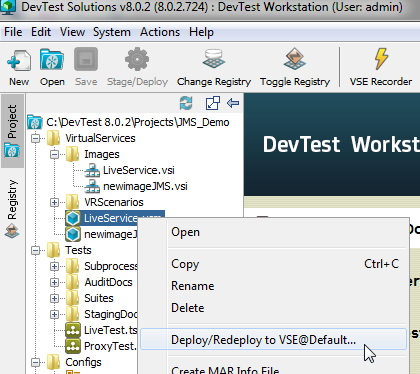
</mbean>

<!-- Added to enablement challenges -->

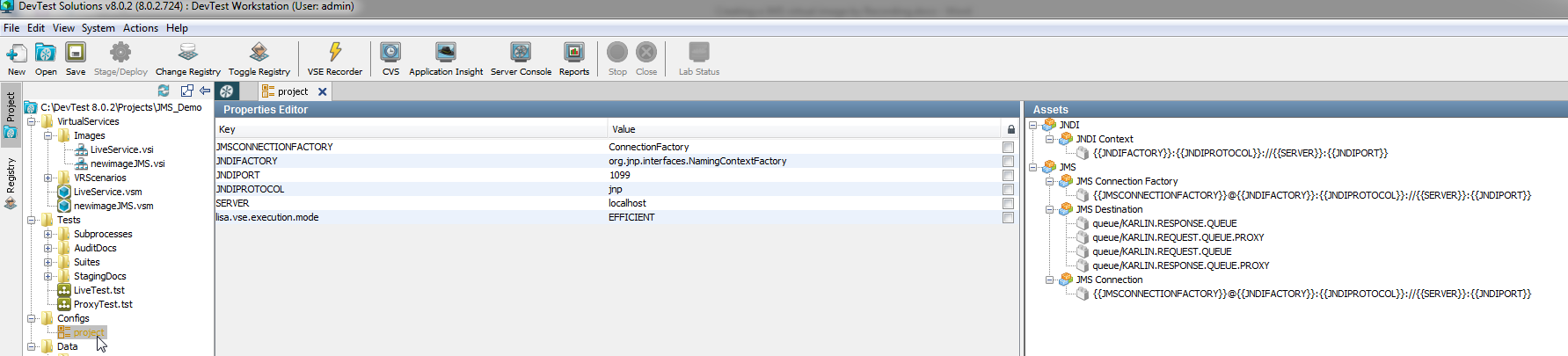
</server>

1. Creating a JMS virtual service by recording and playback:

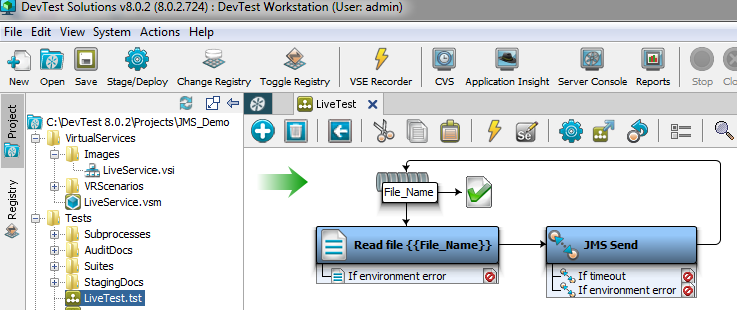
* Invoke the SV DevTest Console in Firefox: <http://localhost:1505>
* Deploy LiveService.vsm to the VSE environment. This is a pre-configured virtual service is in lieu of an actual application server (service) to read the request from the live request queue, sending a response back to the Client containing the data that was queried.



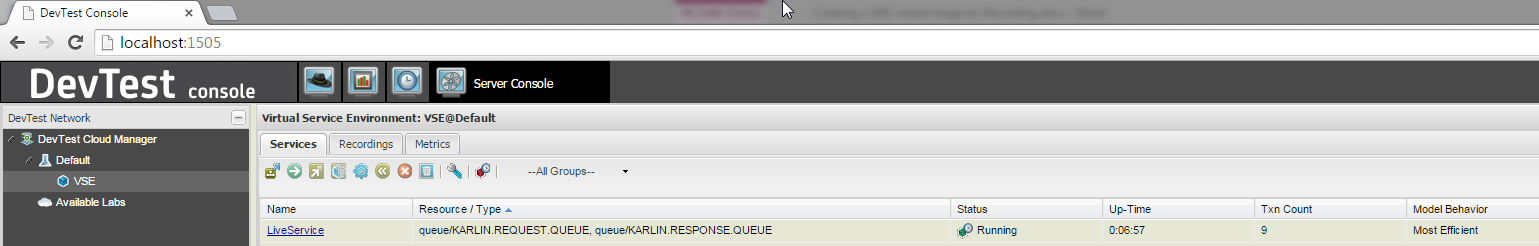
* Open the project configuration file to observe the property editor key and value entries you’ll use to demo. We’ll hit the live queues first to validate the environment is working.



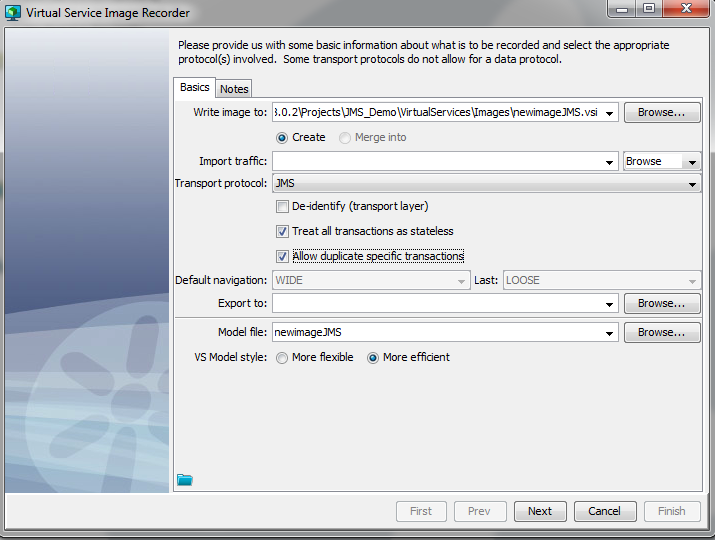
* Open the LiveTest.tst test case and run the test in the ITR
* Invoke the ITR and execute the test



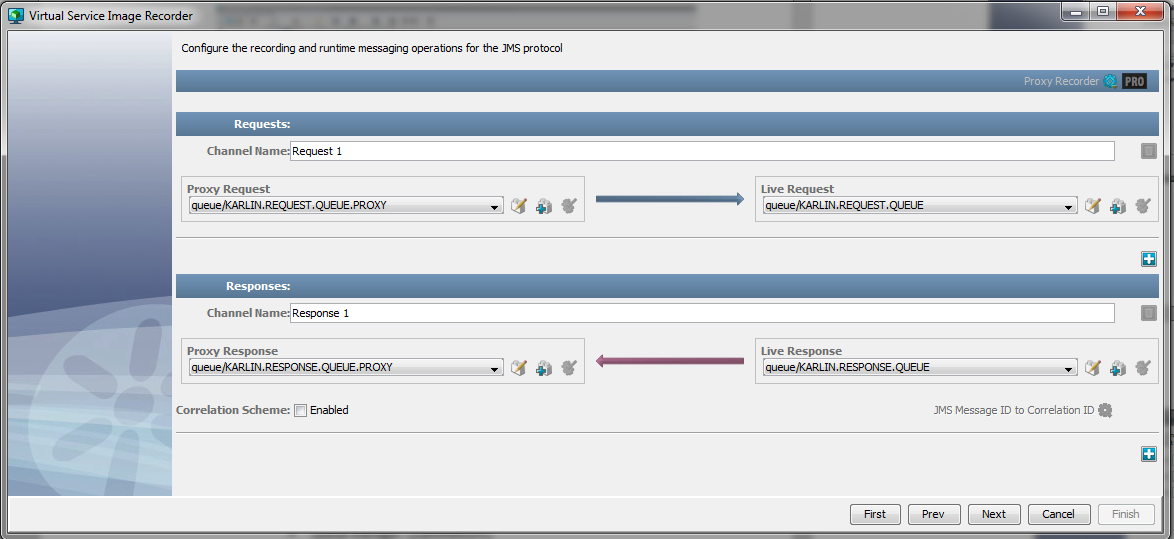
* Confirm 9 transaction against the server-responder model in the DevTest Console

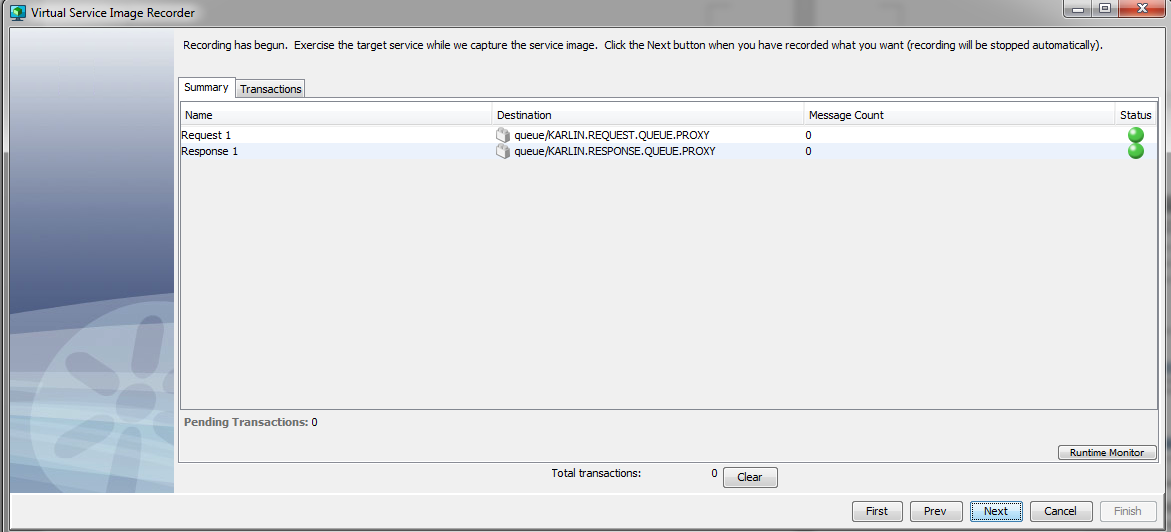


* Reset the server-responder transaction count
* Create a new virtual image by recording
  + newimageJMS.vsi
  + newimageJMS.vsm
  + transport protocol: JMS



* + next: configure the recording and runtime message operations for the JMS Protocol



* + Ensure you have the above drop lists correctly selected
  + Next: Recording has begun.
* Open ProxyTest.tst, invoke the ITR and execute the test

Wait about 20 seconds after the test completes for the counter to catch-up