

Read Me
How to run the Jade Joram Self Optimization Demo ?

Florent Métral - SARDES - INRIA Rhône-Alpes
<http://sardes.inrialpes.fr/jade.html>

July 4, 2008



Contents

1	Prerequisites	3
2	Configuring the Jade Distribution	3
3	Running the platform	4
4	Deploying the infrastructure	4
5	Using Clif to run probes and injectors	5

1 Prerequisites

For this demonstration, you need :

- Jade 2.0.1
- Joram Bundle 1.0.0
- Joram Optimization Bundle 1.0.0
- JoramOptimizationDemo project
- Java 1.5
- Maven 2.0 (or greater)
- Clif console 1.3.1

You can get the Java projects (Jade, bundles sources and demo) on the SVN server *svn.forge.objectweb.org* on the following paths:

- /svnroot/jasmine/jade/tags/JADE_2_0_1
- /svnroot/jasmine/jasmine/tags/JORAM_BUNDLE_1_0_0
- /svnroot/jasmine/jade/tags/JORAM_OPTIMIZATION_BUNDLE_1_0_0
- /svnroot/jasmine/jade/trunk/Managers/JoramOptimizationDemo

2 Configuring the Jade Distribution

In the *system.properties* file under *assemblies/jadenode/src/main/resources/conf* and *assemblies/jadeboot/src/main/resources/conf*, you have to replace localhost by the IP address of your host.

Moreover, in the *pom.xml* file at the root of the Jade distribution, you have to set the *jonathan.connectionfactory.host* system property with your own IP address.

Then, you can compile the Jade distribution by running, in the Jade directory, the command *mvn clean install*.

If you want to launch all the demo on one host, you need to follow these steps :

- Copy the *assemblies/jadenode/target/jadenode* directory as much as you need Jade nodes.
- For each Jade node directory, you need to change the Felix cache directory property. This property must be unique. It can be done by modifying the *felix.cache.profile* property in the *conf/config.properties* file.
- For each Jade node directory, you need to set the *jadeboot.registry.host* and *jadeboot.registry.port* system properties in the *conf/config.properties* file.
- For each Jade node directory, you need to set the *clif.codeserver.host* system property in the *conf/config.properties* file. It must correspond to the host on which you run the clif console.

3 Running the platform

For this demonstration, you need one Jade boot and three Jade nodes.

Launch the Jade boot with the following steps :

- Go in the Jade boot target directory
- Delete the Felix cache directory `rm -rf ~/.felix`
- Delete the tmp folder (Joram uses it to store servers messages used in the demo) `rm -rf /tmp`
- Run the command `sh jadeboot.sh`

Launch two Jade nodes with the following steps :

- Go in the Jade node directory
- Run the command `sh jadenode.sh`

4 Deploying the infrastructure

In this demonstration, we deploy a JNDI registry on the first Jade node and a Joram server hosting a cluster queue composed of one queue (for the moment) on the second Jade node.

To deploy Joram components, we use the FScript console.

In the Jade directory, you can launch it with the following maven command:

- `mvn -P run-fscript`

In the examples/joram directory of Jade, you can find some deployment examples in the deployExamples.txt file.

The one we are going to deploy is the following one :

Cluster Queue on 1 servers - Rigths managed on the cluster queue - Optimisation

```
:rmi_connect localhost:1238
:load examples/joram/joramFelix.fscript
addDomain("JoramDomain_D0");
startJndi("localhost_1","16400");
startManager("localhost_1","1.0");
addServer("localhost_2","JoramDomain_D0","0","true");
addConnectionFactory("0","cf0","true");
addClusterConnectionFactory("ccfClusterQueue_0","true");
addConnectionFactoryToCluster("ccfClusterQueue_0","0","cf0");
addUser("0","anonymous","anonymous");
addUser("0","user","user");
addClusterQueue("ClusterQueue_0");
addQueue("0","queue0","","all","all","yes");
addQueueToCluster("ClusterQueue_0","0","queue0");
```

In the script above, you must change *localhost_1* and *localhost_2* by the 2 Jade nodes names received by the Jade boot. Look at the Jade boot execution traces to get them.

Now, the last point is to start a third Jade node, which will be free of deployments, and so, which will be used to start the new Joram server.

So, start the third Jade node as mentionned above.

5 Using Clif to run probes and injectors

First of all, you have to download the JoramOptimizationDemo project from the Jasmine SVN repository.

Then, you have to launch a Clif console and import this project.

You can get Clif at the following svn location:

svn://svn.forge.objectweb.org/svnroot/clif/tags/v1r3_0

The next step is to connect the Clif console to the Jade boot Fractal RMI registry. In the window where you locate the Jade boot RMI registry, add the JoramOptimizationDemo project to the classpath. This is necessary in order for the Clif console to find the *.xis* files that describe isac scenarios.

To finish, you have to edit the *.ctp* and *.xis* files (using Eclipse Clif perspective) to place the probes and injectors on the Jade nodes. Just deploy, initialize and start the probes and injectors. You'll see a new server with a queue deployed on the free Jade node.