**SysTeam ServiceMix**

# Git Repository:

<https://github.com/systeam-org/CMPE272-HW5.git>

# Setup Instructions:

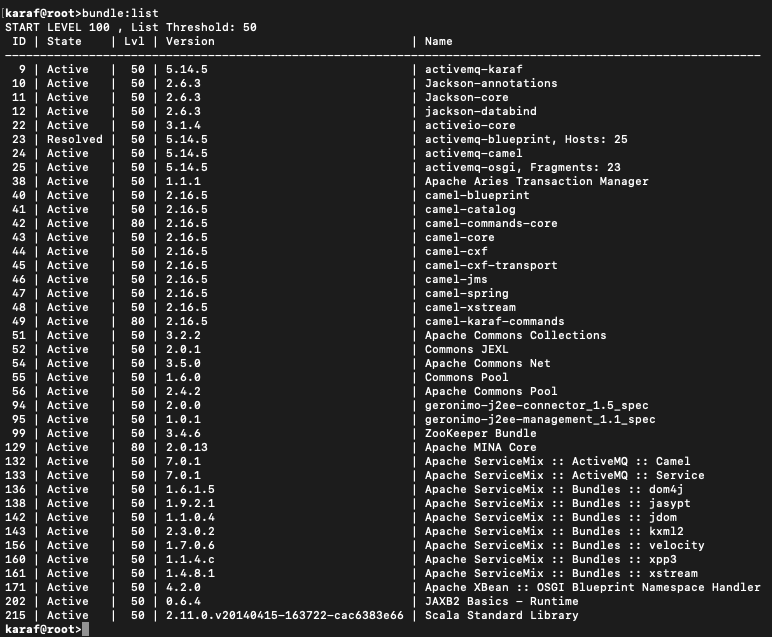
## Installation:

1. Downloaded karaf from <http://servicemix.apache.org/downloads.html> and unzipped it
2. Started servicemix as shown below.

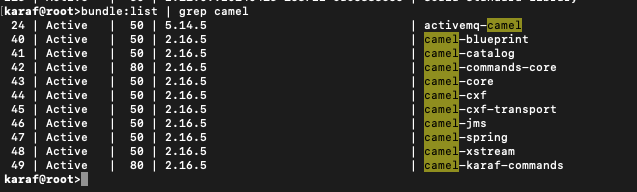


## Apache ServiceMix console

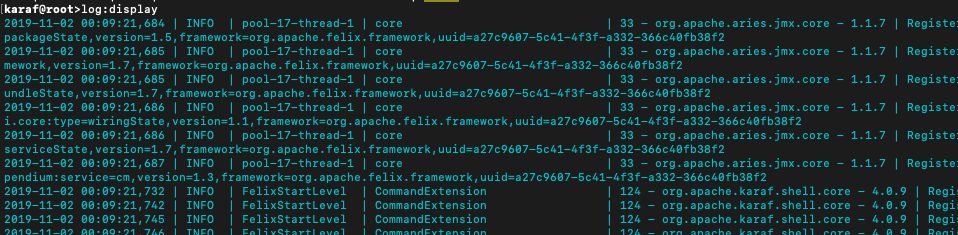
1. We can see the list of currently installed bundles as shown below



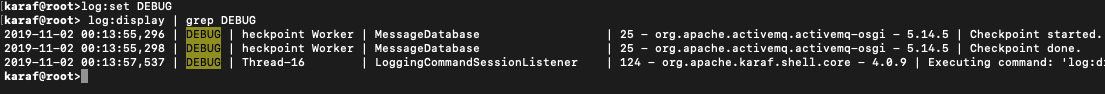
1. We can see camel specific bundles as shown below



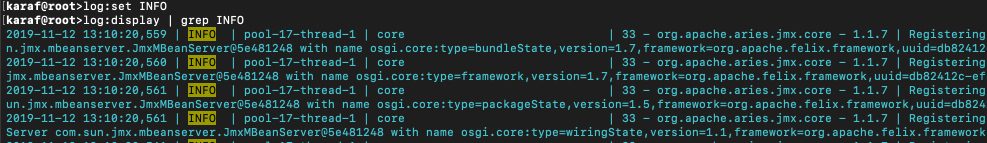
1. We can see the log output as below



1. We can set log level to DEBUG and see corresponding logs in the log file as below



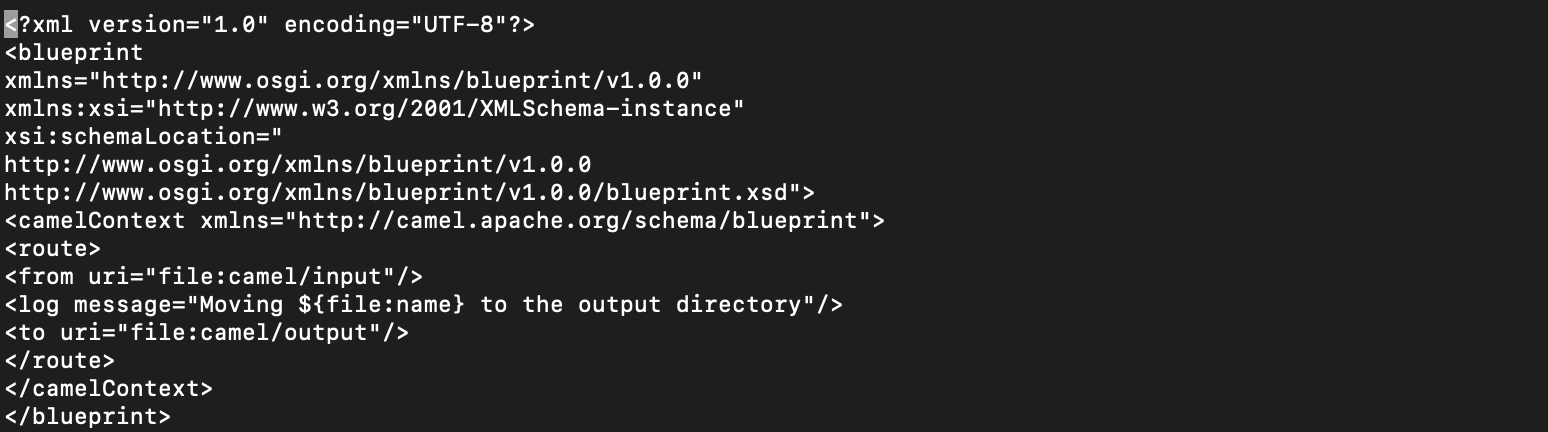
1. We can set log level to INFO and see corresponding logs in the log file as below



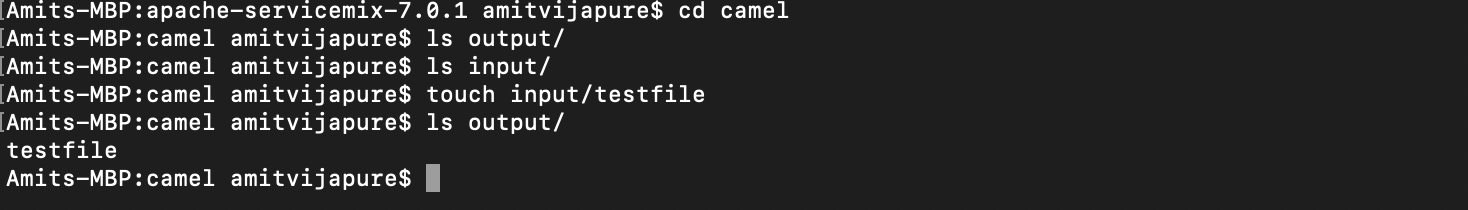
## Using Camel

1. Created ‘blueprint.xml’ file and deployed it in ‘deploy’ folder. This will create route. This route will move all files from ‘camel/input’ folder to ‘camel/output’ folder on creation of any new file in ‘camel/input’ folder.

Contents of blueprint.xml



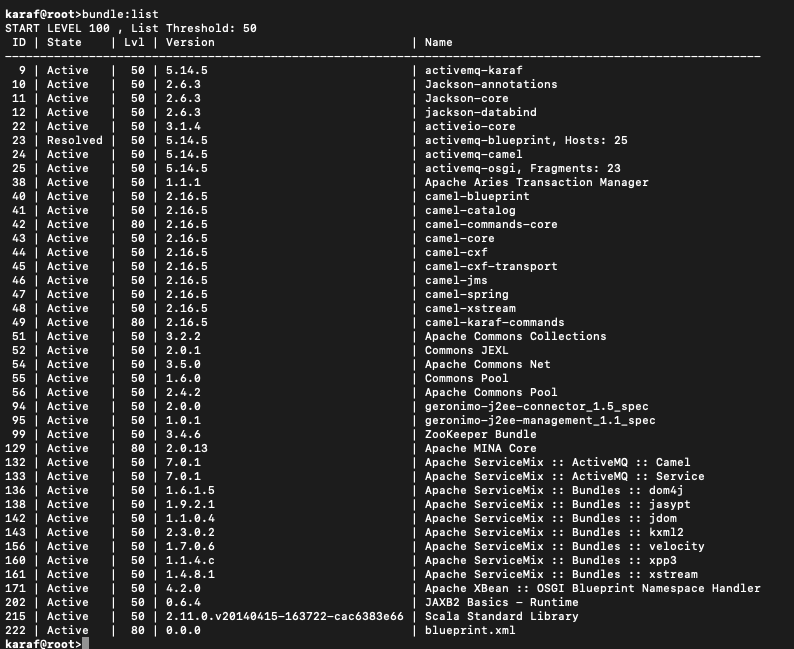
1. Here we created ‘testfile’ in ‘camel/input’ folder. It automatically gets moved to ‘camel/output’ folder.



Here are corresponding logs



1. We can see deployed ‘blueprint.xml’ using ‘bundle:list’ command. We can see ‘blueprint.xml’ at the end of list.



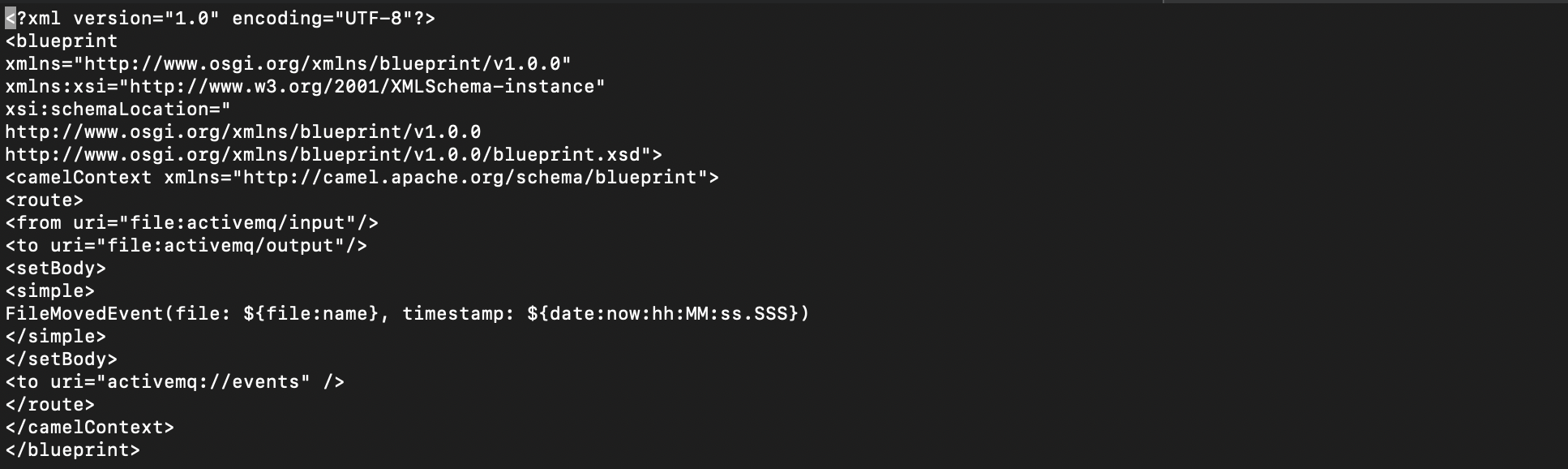
1. We can start/stop as route as follows



## Adding ActiveMQ to the Mix

1. In this step, we created and deployed ‘activemq.xml’ file in ‘deploy’ folder. This route will move all new created files in ‘activemq/input’ folder to ‘activemq/output’ folder. And here you will not see any events in log file. Instead they are sent to ActiveMQ queue called ‘events’.

Contents of activemq.xml



1. Now we created ‘testfile’ in ‘activemq/input’ folder and it gets automatically moved to ‘activemq/output’ folder. Corresponding info sent to ‘events’ queue instead of logs.

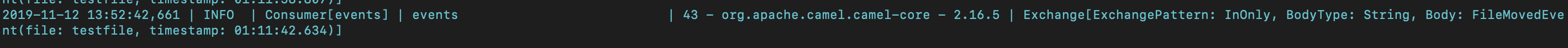


1. To retrieve the info sent to ‘events’ queue in last step, we created and deployed another blueprint ‘activemq\_receive\_events.xml’ in ‘deploy’ folder. Immediately after deployment, it will receive info from ‘events’ queue and get added it to logs.

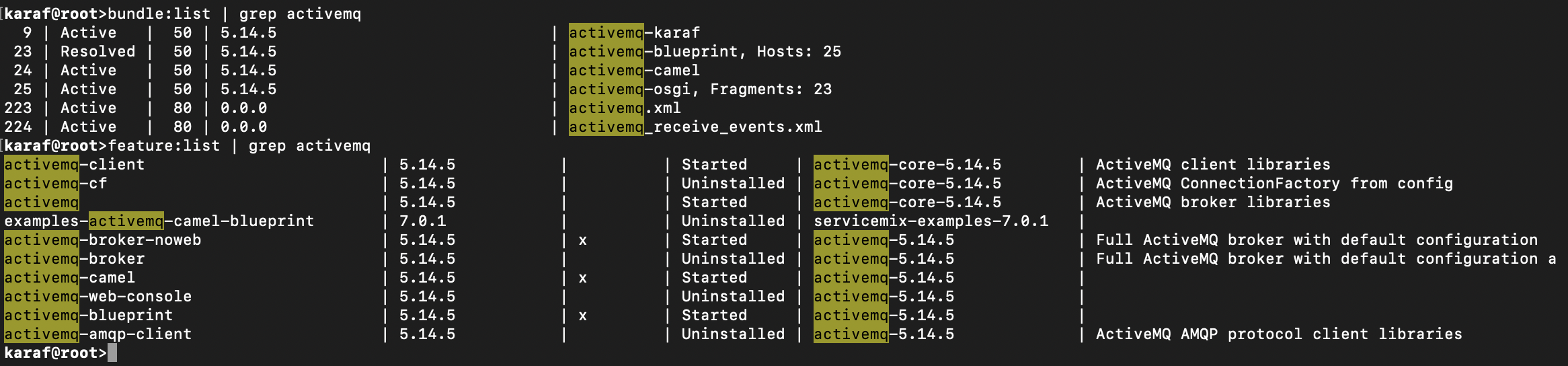
Contents of ‘activmq\_receive\_events.xml’



1. After deployment, we can see below logs in in the log file which are received from ‘events’ queue.



1. We can see ActiveMQ related bundles and features as follows:



## Simple JMS application multiple, concurrent, consumers and producers

Consumer1

Producer1

Consumer2

Producer2

Consumer10

Producer10

In this application, we have 10 concurrent producers and 10 concurrent consumers for on queue ‘ESP’ and they are using ActiveMQ in karaf.

Our code located in: <https://github.com/systeam-org/CMPE272-HW5.git>

To run this program, we need activemq\_all.jar in classpath which can be downloaded from <https://mvnrepository.com/artifact/org.apache.activemq/activemq-all/5.15.9>

So, we added this jar as External Jar in our classpath in Eclipse and run our producer consumer program.

Output:



There are 10 producers and 10 consumers. We can see that, message produced by Thread-1 is received by Thread-2. And there is similar producer-consumer thread mapping like Thread-0 to Thread-7, Thread-8 to Thread-5, Thread-6 to Thread-11, etc.