

# Enterprise Systems & Architecture

## Lab 10 (Weeks 10-12): JMS Publish & Subscribe

*Note: Complete over the remaining lab sessions.*

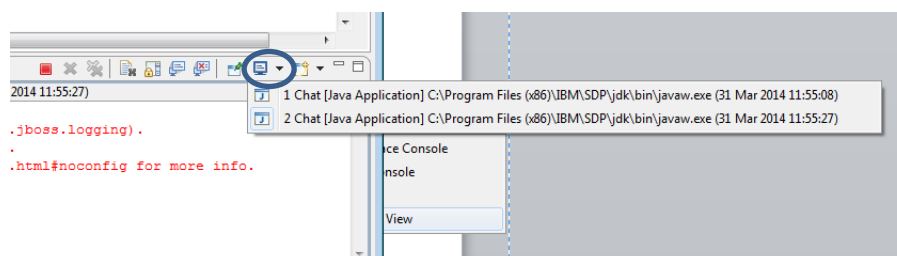
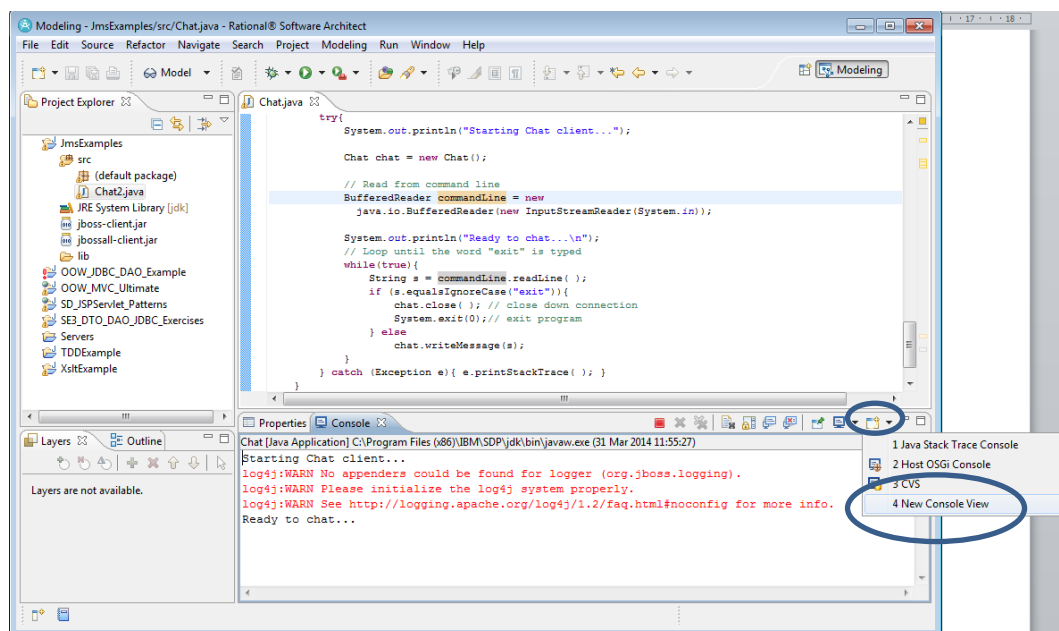
**In this lab you can use the Java project you created and worked on in the last lab.**

### Demonstration of Publish & Subscribe Messaging using a Simplified Chat Application

- Using file explorer, copy the *Chat.java* source file from the Webcourses lab folder into your project *src* folder.
- Study the code to see how it uses the Publish / Subscribe mechanism to send messages between instances of the program. Note the *InputStreamReader*, when your code is running, you can enter text using the *Console View*.
- Ensure you have started the JBoss Application Server as in the last lab.
- From eclipse, run the code twice (or three times). Ignore the warnings for now. Please see next note.

#### Note:

When you run two (or three) instances of the code, they will be running simultaneously and will keep running until the user types “exit”. In eclipse, you will probably only see one *Console View*. You can open multiple *Console* views and also select which running process uses each *Console*. You can also *Detach* views and arrange them side by side if you want (see below). In this way you can test out the messages being sent and received by each instance of your code. Please ask if you need help with the *Console* views.



## Exercise 1

- Test out the chat clients. Note that when a message is sent by one client, all clients receive that message (including the one that sent it).
- Modify the code so that the client that sent the message does not print it out on the console when it receives it.

**Hint:** Have a look at the methods on the JMS Message Interface specification (link below). In JMS, you can set properties into the message before sending them and read them from the message after receiving them. See if you can use some of the methods to achieve the above.

<http://docs.oracle.com/javaee/6/api/javax/jms/Message.html>

## Exercise 2

- Modify the code so that the user could enter a name at the Console which would subsequently be used to prefix any messages sent by the client (e.g. *Ciaran> message text...*).

**Suggestion:** With the existing code, the user can enter “exit” to shut down the client. In a similar way, the user could enter “myname=Ciaran” to tell the client to prefix any messages with that name.