

JMS „Hello World“ Project

This document shows how to open and run the JMS “Hello World” project using the Eclipse IDE.

It is assumed that Java (version 7 or later) is already installed on your system. This document assumes a Windows system, but everything should also work on Linux.

The following tools will be used:

- Eclipse
An IDE (Integrated Development Environment) for Java (and many other languages), that also supports Maven (see below).
- Apache ActiveMQ
A Messaging middleware that offers a JMS API.
- Apache Maven
A build management tool. The “Hello World” project is provided as a Maven project, i.e. you can use any IDE that supports Maven to build and run the project. This documentation however assumes that you are using Eclipse.

1 Eclipse IDE

Download the Eclipse IDE from:

<http://www.eclipse.org/downloads/eclipse-packages/>

Choose the “Eclipse IDE for Java EE Developers” package.

Installing and running Eclipse is easy, just unzip the downloaded file and then run “eclipse.exe”.

2 ActiveMQ

Download ActiveMQ from:

<http://activemq.apache.org/download.html>

Unzip the file, go to the “bin” folder and run ActiveMQ from the command line with “activemq start”.

Open <http://127.0.0.1:8161/admin/> and log in using admin/admin as username/password. More information can be found here:

<http://activemq.apache.org/getting-started.html>

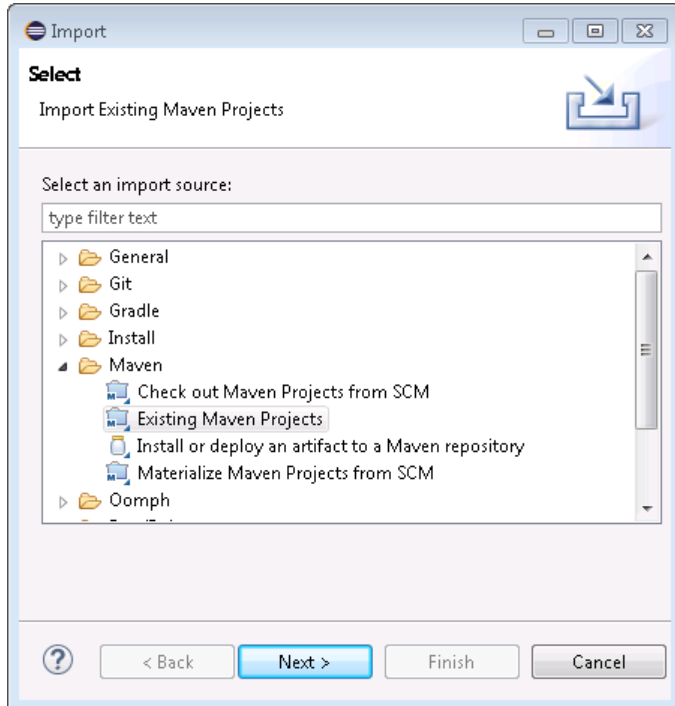
3 “Hello World” project

3.1 Import to Eclipse

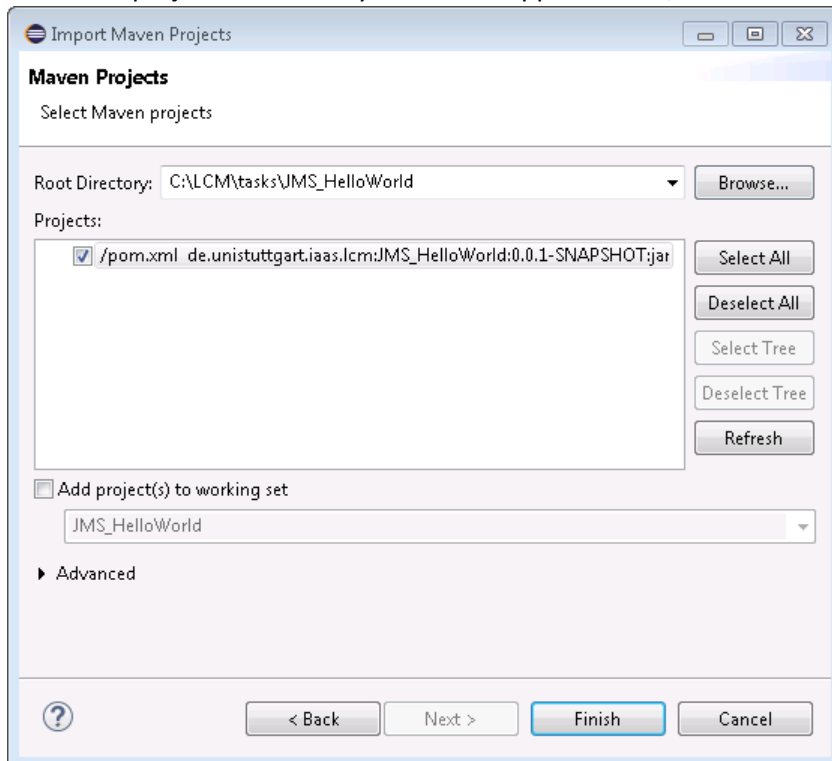
Download and unzip the project. Then open Eclipse and import the project as follows:

- File → Import...

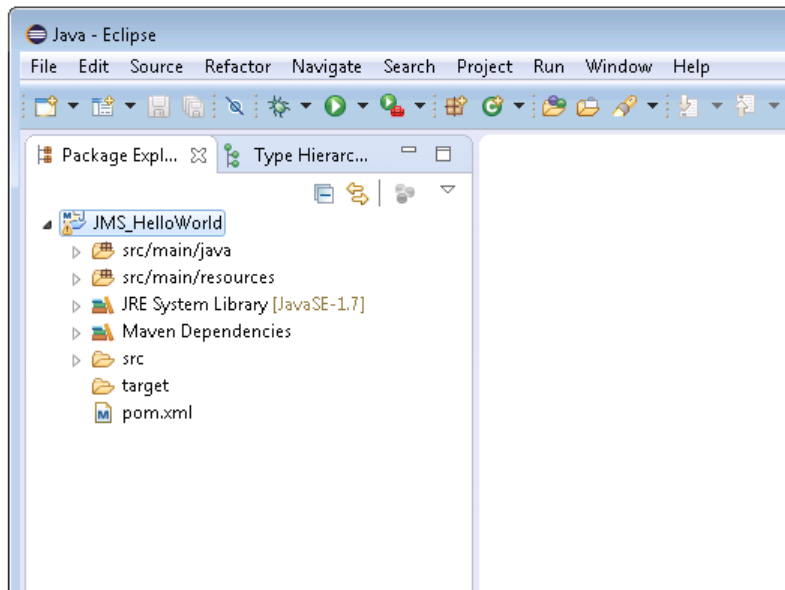
- Choose “Existing Maven Projects” from the “Maven” category



- Select the project folder that you have unzipped before, then select “Finish”

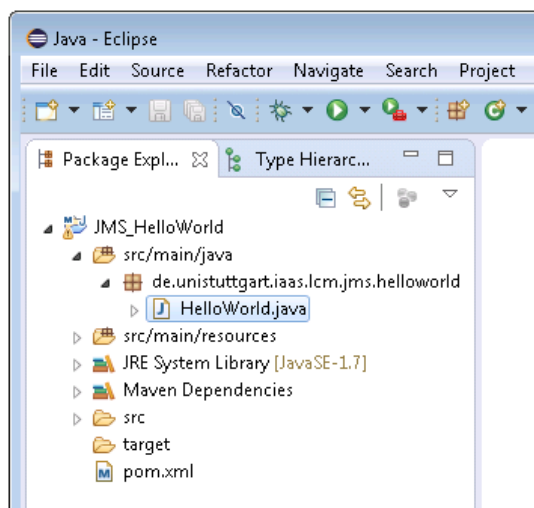


- The project should then appear in your workspace

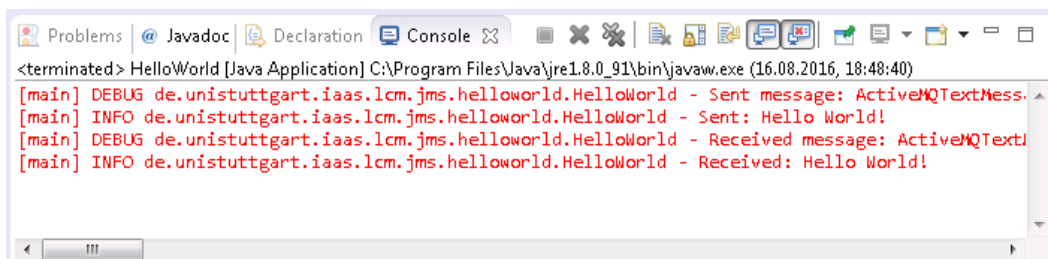


3.2 Running the project

Make sure that ActiveMQ is running (see section 2). In Eclipse, navigate to “HelloWorld.java”, right-click on it, and then select “Run As → Java Application” from the context menu.



If everything works as expected, the console output in Eclipse should look like this:



3.3 Understanding the project structure

All application logic is contained in the “HelloWorld” class. The example can be run using the main method of the “HelloWorld” class. The program first connects to ActiveMQ (using JNDI and JMS), then sends a text message to a queue, and finally reads the text message from the same queue.

Instead of using a fully-fledged JNDI server, all objects that should be available using JNDI are configured in the “jndi.properties” file located in the “src/main/resources” subfolder.

Topics and Queues don’t have to be created in advance. ActiveMQ creates them “on the fly” when they are used the first time.

The project uses SLF4J as log library, connected to the simple logger library. Logging can be configured through the “simplelogger.properties” file located in the “src/main/resources” subfolder.