|  |
| --- |
|  |
| Developer Installation Guide |
| Online decision support toolkit for climate resilient seaports |

|  |
| --- |
| **RMIT University**  Prepared by: **Guillaume Prevost**  Date: **03/06/2013**  ANDS Project Code: **AP35**  Document Version: **1.0** |

|  |  |
| --- | --- |
| **ANDS Project Code** | AP35 |
| **Project Title** | Online decision support toolkit for climate resilient seaports |
| **ANDS Program** | Applications |
| **Organisation responsible for the project (Subcontractor)** | RMIT University |
| **Organisation that will undertake the work (Sub-Subcontractor)** | RMIT University |
| **Name of Contact Person** | Professor Darryn McEvoy |
| **Address and contact details of Contact Person** | Address: GPO Box 2476, Melbourne VIC 3001  Telephone: 03 99251943  Email: [Darryn.mcevoy@rmit.edu.au](mailto:Darryn.mcevoy@rmit.edu.au) |
| **Names and affiliations of collaborators if any** | Climate Change Adaptation Program and eResearch at RMIT University, Ports Australia, National Transport Commission |

# License

*Copyright (c) 2013, RMIT University, Australia.*

*All rights reserved.*

*This project’s code and documentation are under the BSD license. See 'license.txt' for details.*

*Project hosted at:* [*https://code.google.com/p/climate-smart-seaports*](https://code.google.com/p/climate-smart-seaports)

Table of Contents

[Document Revision History 3](#_Toc357690248)

[1. Introduction 4](#_Toc357690249)

[1.1. Purpose 4](#_Toc357690250)

[1.2. Intended Audience and Reading Suggestions 4](#_Toc357690251)

[1.3. References **Error! Bookmark not defined.**](#_Toc357690252)

[2. Prerequisites 5](#_Toc357690253)

[2.1. List of prerequisites 5](#_Toc357690254)

[2.2. Java Development Kit 1.7 5](#_Toc357690255)

[2.3. MAVEN 5](#_Toc357690256)

[2.4. Eclipse 6](#_Toc357690257)

[2.5. Eclipse Plugin: Subclipse 6](#_Toc357690258)

[2.6. Eclipse Plugin: m2e 6](#_Toc357690259)

[2.7. Eclipse Plugin: Mylyn 7](#_Toc357690260)

[2.8. Eclipse Plugin: Run Jetty Run 7](#_Toc357690261)

[2.9. Eclipse Plugin: Maven Integration for WTP 8](#_Toc357690262)

[2.10. MySQL 8](#_Toc357690263)

[3. Installation **Error! Bookmark not defined.**](#_Toc357690264)

[3.1. Connection Eclipse-MySQL 9](#_Toc357690265)

[3.2. Retrieve project from SVN 17](#_Toc357690266)

[3.3. Troubleshoot project issues 22](#_Toc357690267)

[3.3.1. Select the right jre System Library 22](#_Toc357690268)

[3.3.2. Select the right Run configuration 24](#_Toc357690269)

[3.3.3. File upload issues at runtime 25](#_Toc357690270)

[4. Deployment under UNIX 26](#_Toc357690271)

[4.1. Pre-requisites 26](#_Toc357690272)

[4.2. Deployment **Error! Bookmark not defined.**](#_Toc357690273)

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Date | Modifications | Version |
| Guillaume Prevost | 10/12/2012 | First section regarding prerequisite installation. | 0.1 |
| Guillaume Prevost | 11/12/2012 | Addition of section 2.8 | 0.2 |
| Guillaume Prevost | 12/12/2012 | Formatting change | 0.3 |
| Guillaume Prevost | 13/12/2012 | Addition to the MySQL Configuration details  Addition of the SVN Configuration details | 0.4 |
| Guillaume Prevost | 21/12/2012 | Addition of the project deployment troubleshoots | 0.5 |
| Guillaume Prevost | 22/01/2013 | Renaming of the file and the title hierarchy | 0.6 |
| Guillaume Prevost | 30/05/2013 | Completed the application deployment section | 0.7 |
| Guillaume Prevost | 03/06/2013 | Completed the application UNIX pre-req. section | 1.0 |

# Introduction

## Project Summary

The potential impact of climate change on ports differs according to their location, function and business model of the ports. To be 'Climate Smart', ports in Australia need to understand the relevant climate impacts and risks for their particular operation; only then can they determine what adaptation measures may be appropriate.

The Climate Smart Seaports web-tool is designed primarily for port personnel who make (or influence) decisions around long-term port planning for infrastructure, assets and management systems. However, it will also be of value to port owners and related businesses, government departments and local authorities concerned with ports and infrastructure; and for application by academic researchers.

The Climate Smart Seaports Tool enables interested users to begin the process of a climate risk assessment. It assists them to identify current and historical climate trends and variability, as well as future climate projections under a variety of scenarios.

Population and trade data is included, and users can add port-specific information to round out their analysis.

## Purpose

The purpose of this document is to detail the prerequisite and the steps to set up the development environment for the Seaports project.

## Intended Audience and Reading Suggestions

This document is aimed to new developers starting to work on the project. It details the prerequisites to install, the initial configuration to prepare the system for development the project, how to retrieve the project sources and build the project.

# Prerequisites

## List of prerequisites

|  |  |
| --- | --- |
| Prerequisite Software | Detailed in Section |
| Java JDK 7 | 2.2 |
| Maven | 2.3 |
| Eclipse Helios | 2.4MAVEN**Error! Reference source not found.** |
| Eclipse plugin: Subclipse | 2.5 |
| Eclipse plugin: M2E | 2.6 |
| Eclipse plugin: Mylyn | 2.7 |
| Eclipse plugin: Run Jetty Run | 2.8 |
| MySQL | 2.10 |

## Java Development Kit 1.7

This is the Java Development Kit, it is needed to get the required Java development tools such as the java compiler available. When you install the JDK, it automatically installs the Java Runtime Environment 7, necessary to run Java applications such as Eclipse.

Required version: **JDK 1.7.0\_09**

Download at: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Install the downloaded file by following the installer step.

## MAVEN

Maven will be used to manage the project’s dependencies and deploy the project.

Required version: **Maven 3.0.5**

Download at: <http://maven.apache.org/download.cgi>

Follow the Maven tutorial at the following address:

<http://www.mkyong.com/maven/how-to-install-maven-in-windows/>

## Eclipse

Eclipse is the IDE used to develop the solution.

Required version: **Eclipse Helios for Java EE developers Service Release 2**

Download at: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/heliossr2>

Simply decompress the downloaded ZIP archive and copy the “eclipse” folder to your application folder (ex: under “c:\Program Files” under Windows).

## Eclipse Plugin: Subclipse

This plugin is required to be able to Update and Commit the changes on the source code from and to the SVN repository.

Version: **1.8.16**

Website: <http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA>

Eclipse repository: <http://subclipse.tigris.org/update_1.8.x>

The following features should be installed:

|  |  |
| --- | --- |
| Package Name | Version |
| CollabNet Merge Client | 3.0.11 |
| JNA Library | 3.4.0.t20120117\_1605 |
| Subclipse | 1.8.16 |
| Subclipse Integration for Mylyn 3.x | 3.0.0 |
| Subversion Client Adapter | 1.8.3 |
| Subversion JavaHL Native Library Adapter | 1.7.6 |
| Subversion Revision Graph | 1.1.1 |
| SVNKit Client Adapter | 1.7.5.1 |
| SVNKit Library | 1.7.5.v1\_r9382\_v20120718\_1415 |

## Eclipse Plugin: m2e

This plugin integrates Maven into Eclipse. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Version: **1.2.0**

Website: <http://www.eclipse.org/m2e/download/>

Eclipse repository: <http://download.eclipse.org/technology/m2e/releases>

The following features should be installed:

|  |  |
| --- | --- |
| Package Name | Version |
| m2e – Maven Integration for Eclipse | 1.2.0.20120903-1050 |
| m2e – slj4j over logback logging | 1.2.0.20120903-1050 |

## Eclipse Plugin: Mylyn

This plugin is required integrate Mylyn to Eclipse. Mylyn is a task and application lifecycle management (ALM) framework for Eclipse. It provides: a task-focused interface, a task management tool for developers.

Version: **3.8.2**

Website: <http://www.eclipse.org/mylyn/downloads/>

Eclipse repository: <http://download.eclipse.org/mylyn/releases/latest>

The following features should be installed:

|  |  |
| --- | --- |
| Package Name | Version |
| Mylyn Context Connector: Eclipse IDE | 3.8.2.v20120916-1200 |
| Mylyn Context Connector | 3.8.2.v20120916-1200 |
| Mylyn Task List | 3.8.2.v20120916-1200 |
| Mylyn Task-Focused Interface | 3.8.2.v20120916-1200 |
| Mylyn Tasks Connector: Bugzilla | 3.8.2.v20120916-1200 |

## Eclipse Plugin: Run Jetty Run

This plugins allows running web applications with Jetty and Eclipse. It features full maven support, running a maven J2EE project and avoids any configuration.

Version: **1.3.3**

Website: <http://code.google.com/p/run-jetty-run/>

Eclipse repository: <http://run-jetty-run.googlecode.com/svn/trunk/updatesite>

The following features should be installed:

|  |  |
| --- | --- |
| Package Name | Version |
| Run Jetty Run Feature | 1.3.3.201203161919 |
| RunJettyRun Jetty 7 Support | 1.3.3.201203161919 |
| RunJettyRun Jetty 8 Support | 1.3.3.201203161919 |

## Eclipse Plugin: Maven Integration for WTP

This plugins allows running web applications with Jetty and Eclipse. It features full maven support, running a maven J2EE project and avoids any configuration.

Version: **0.16**

Website: <http://marketplace.eclipse.org/node/441371>

Eclipse repository: <http://m2eclipse.sonatype.org/sites/m2e-extras>

The following features should be installed:

|  |  |
| --- | --- |
| Package Name | Version |
| Maven Integration for Eclipse WTP (Incubation) | 0.16.0.20120914-0945 |
| m2e connector for mavenarchiver pom properties | 0.15.0.201207090125-signed-201209140800 |

## MySQL

Under Windows, it is possible to download and run the MySQL Installer which will automatically install all the required features listed below: <http://dev.mysql.com/downloads/installer/>

MySQL Community Server is the type of database chosen for the project. In production, a server hosts the MySQL database, but for to ease the development and debug, it is better to Install MySQL on the development machine to host a local database.

Version: **5.5.28**

Website: <http://dev.mysql.com/downloads/mysql/>

MySQL Workbench is a Graphic User Interface for MySQL. It is strongly recommended to install it to avoid managing MySQL from the command line.

Website: <http://dev.mysql.com/downloads/connector/j/>

Version: **5.2.44**

MySQL Connector J is the connector between MySQL and Java. It is required to install it in order to use MySQL with Java.

Version: **5.1.22**

Website: <http://dev.mysql.com/downloads/workbench/5.2.html>

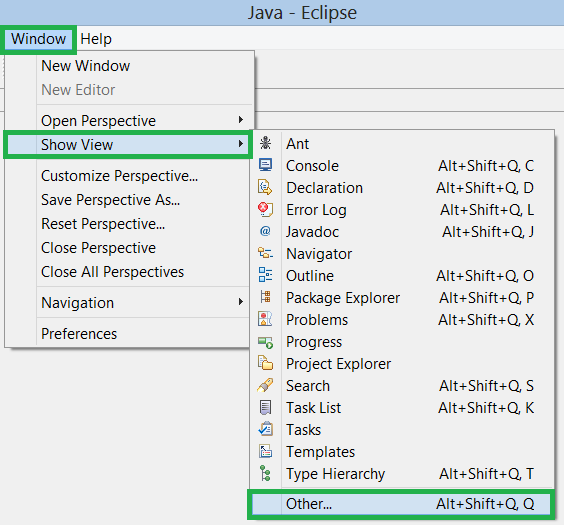
Once installed, proceed to the **configuration of a new MySQL instance**. Under Windows, this is done through a wizard. Follow the steps and specify when required an instance name, a root login and password, and a new user login and password.

The connection with the Eclipse IDE is detailed step by step in section 3.1.

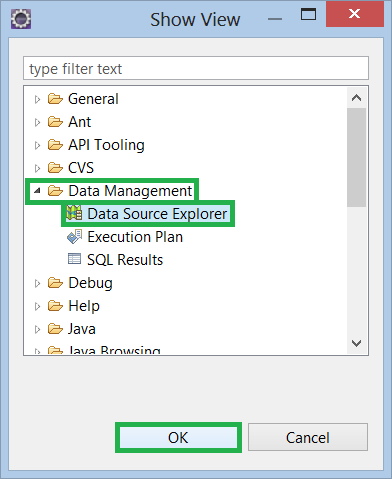
# Setup project for Development

## Connection Eclipse-MySQL

This section explains how to create a connection to a MySQL Database within Eclipse. To start, run Eclipse.

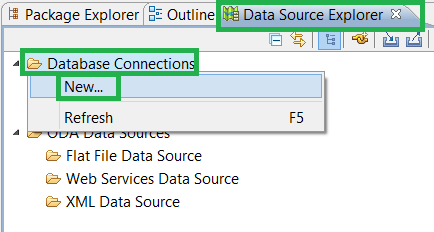


Click on “Window”, select “Show View” and click on “Other…”

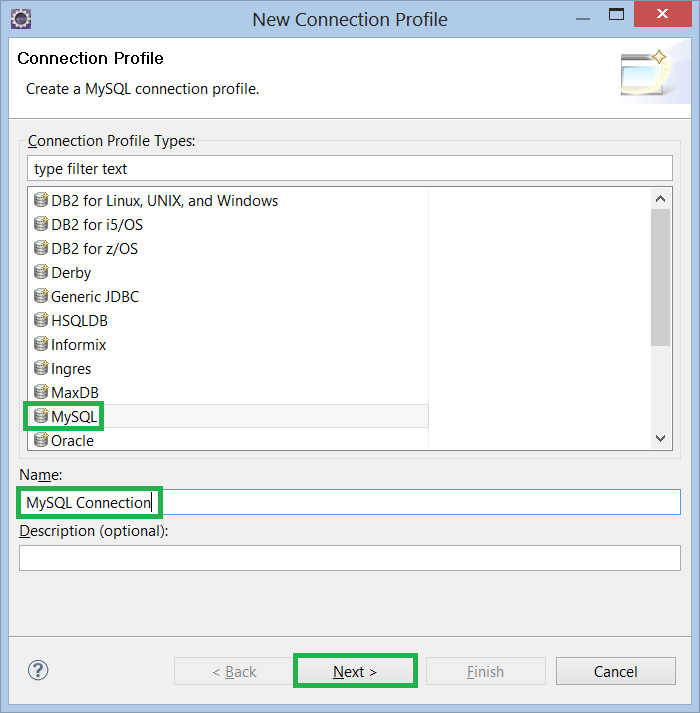


Unfold the “Data Management” folder and select the “Data Source Explorer View”. Click “Ok”.

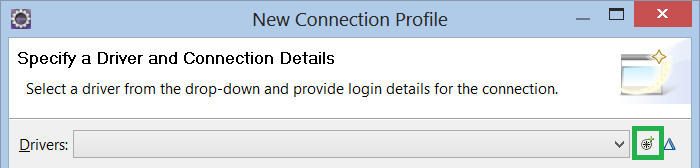
This opens the “Data Source Explorer” view in Eclipse.



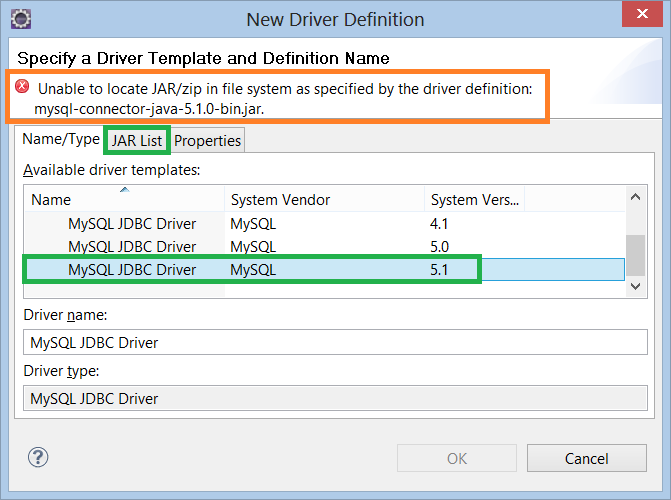
I this view, right click on “Database Connection”, and click on “New…”.



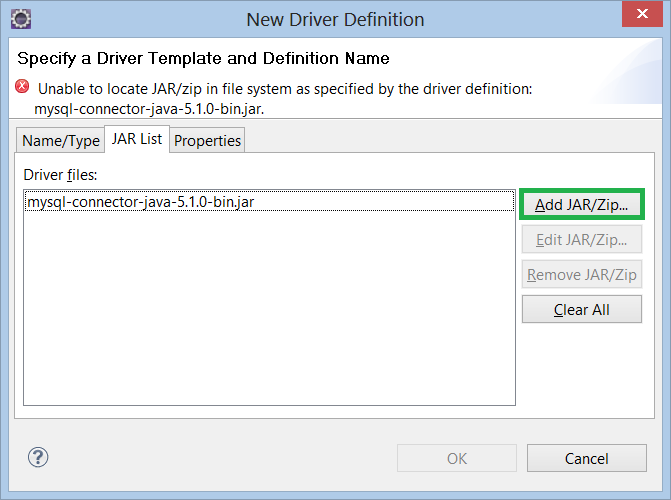
In the list of connection types, select “MySQL”. Type a name for the connection between Eclipse and MySQL that you are going to create. Click “Next”.



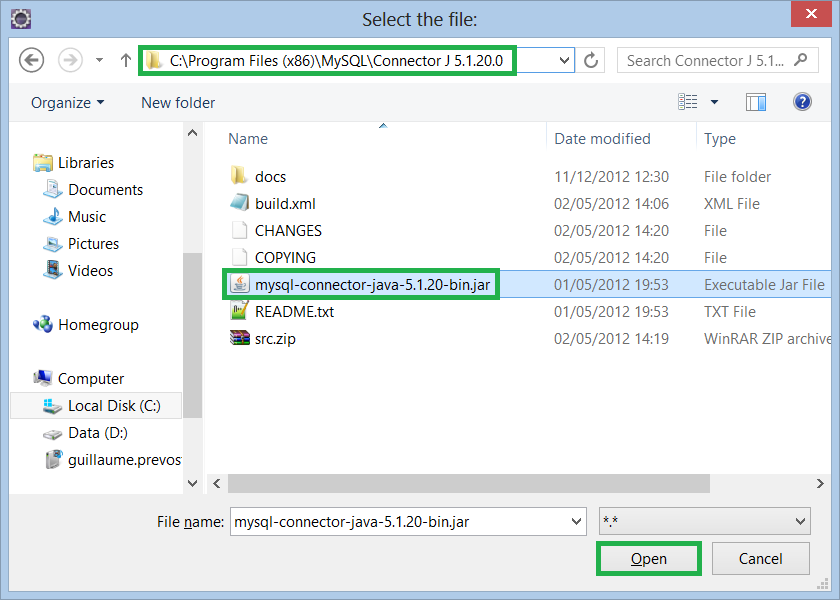
Click on the icon on the right of the empty driver list to create a new driver.



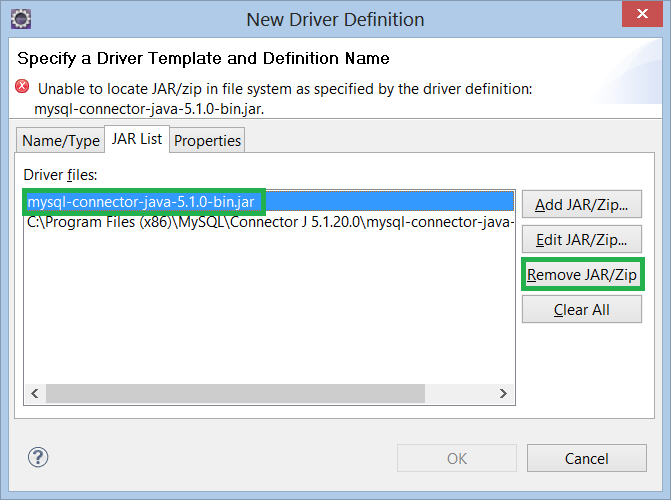
Select the latest MySQL JDBC driver available in the list. If the error message “Unable to locate JAR/zip in file system as specified by the driver definition: …” is displayed, click on the “JAR List” tab and follow the instructions below:



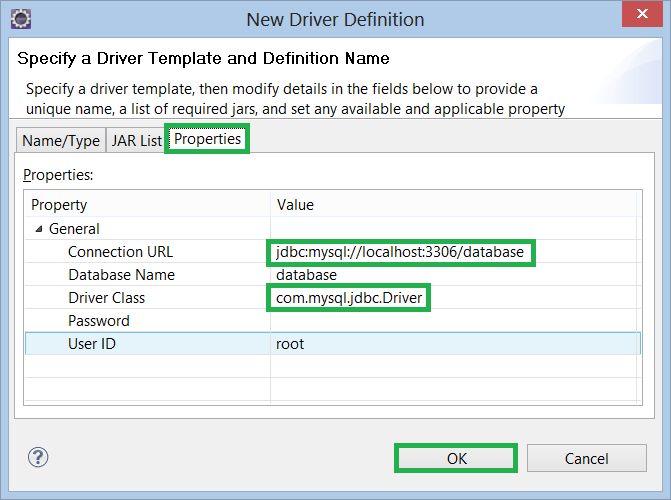
In the “JAR List” tab, click on the “Add JAR/zip…” button.



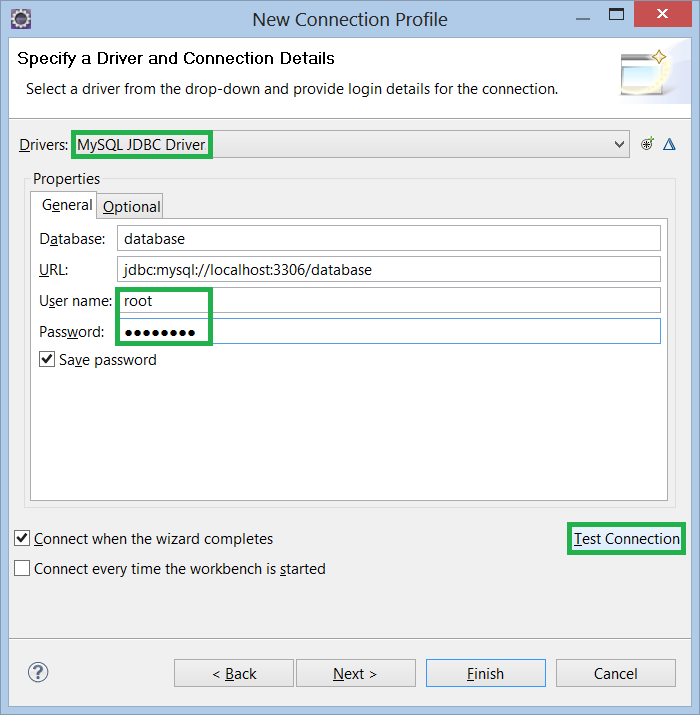
Browse to the installation folder of “MySQL connector J”. By default, it is located in “C:\Program Files (x86)\MySQL\Connector J x.x.x.x” under Windows. Select “mysqlconnector-java-x.x.x-bin.jar” and click “Open”.



Once the new JAR file has been added, select the JAR file which causes the error and click “Remove JAR/zip”.

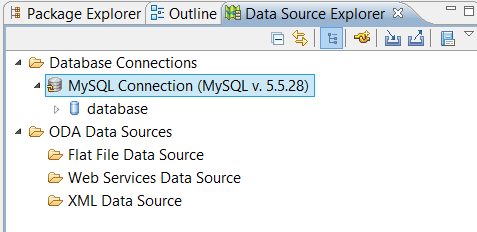


Click on the “Properties” to check whether all the information is correct, and click “OK”.



The Driver has been created and is selected in the driver list. Enter the credentials to connect to the database. Click on “Test Connection” and a message box should appear, saying “Ping succeeded!” if everything was properly configured.

Click “Finish”.

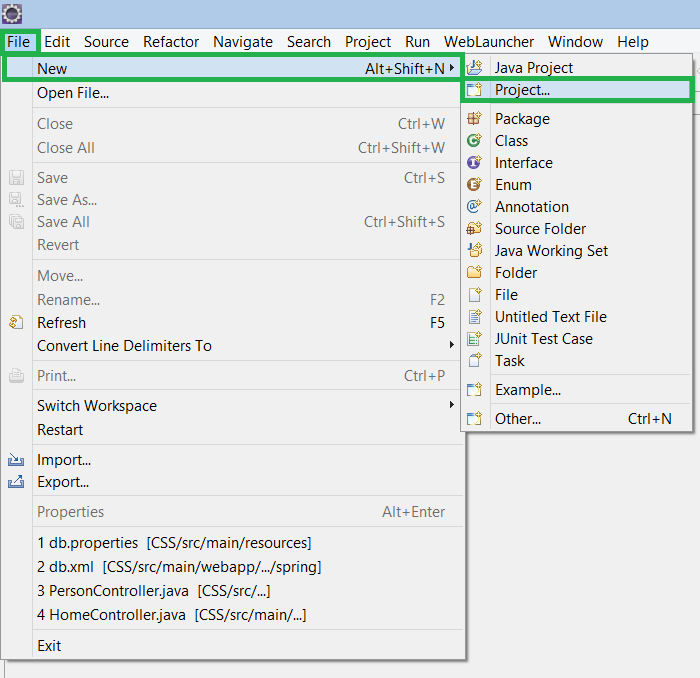


The MySQL connection should now appear in the “Data Source Explorer” view in Eclipse and it is possible to explore the database from there.

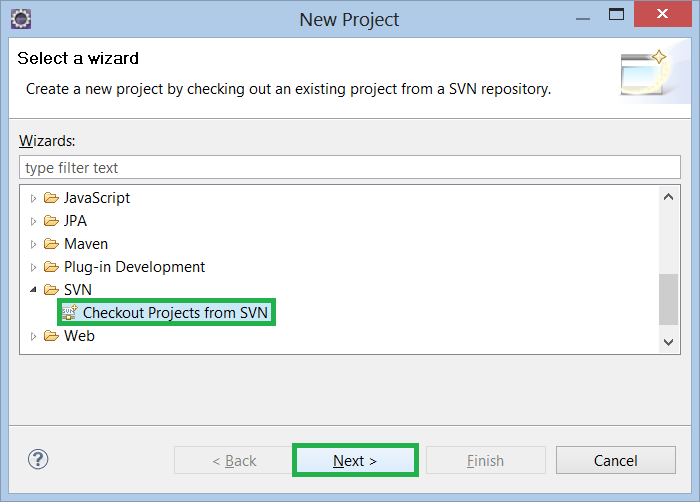
## Retrieve project from SVN

The project sources are located on a SVN repository. To retrieve them directly under Eclipse and be able to update and commit changes directly from the IDE, follow the steps below:

Run Eclipse.

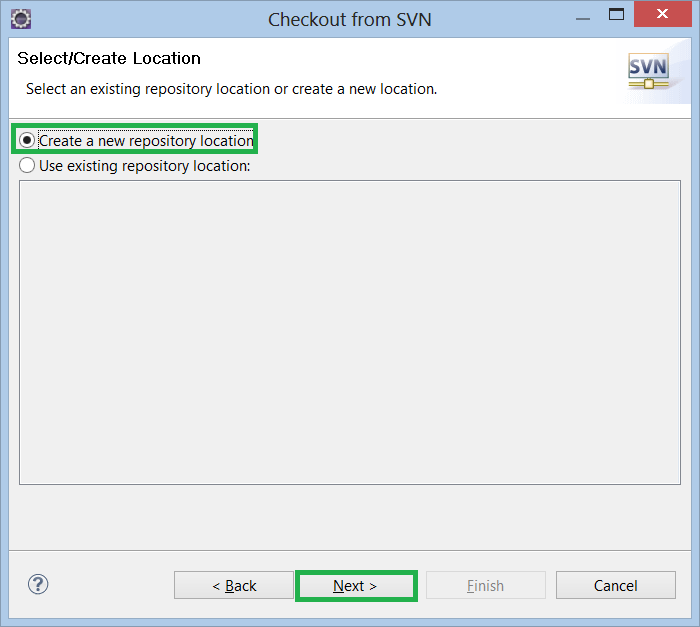


Click on “File”, select “New” and Click on “Project…”

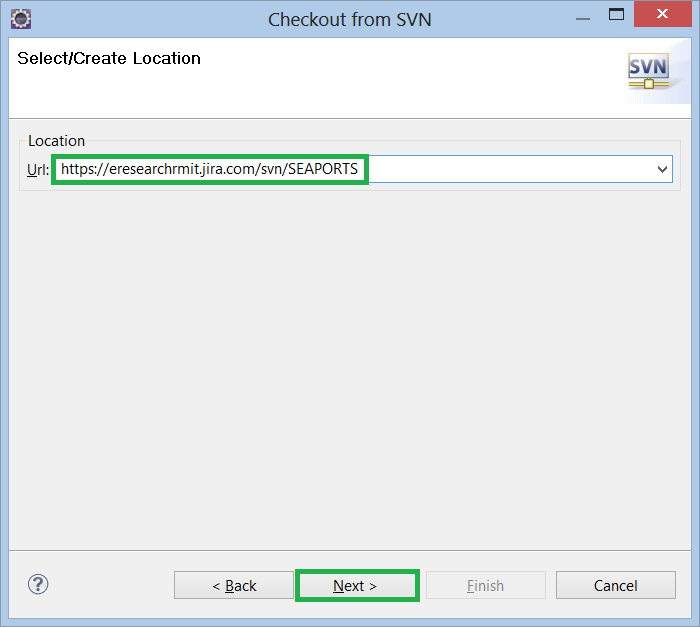


In the list on project types, unfold “SVN” and select “Checkout project from SVN”.

Click “Next”.

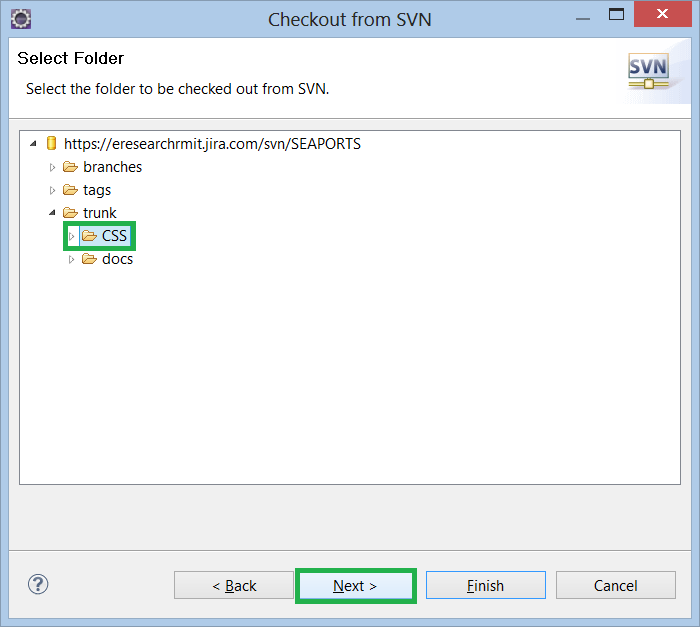


Select the option “Create a new repository location” and Click “Next”

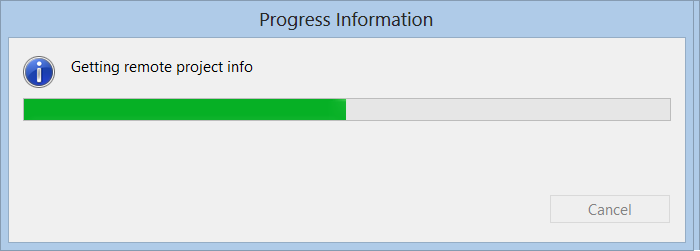


Type the URL of the SVN repository: <https://eresearchrmit.jira.com/svn/SEAPORTS>

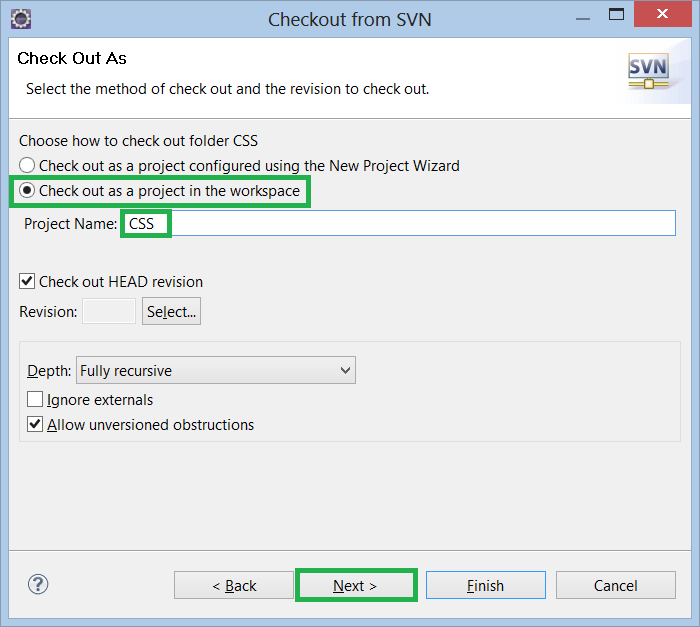
Click “Next”.



Unfold the “trunk” folder and select the “CSS” folder, then click on “Next”.



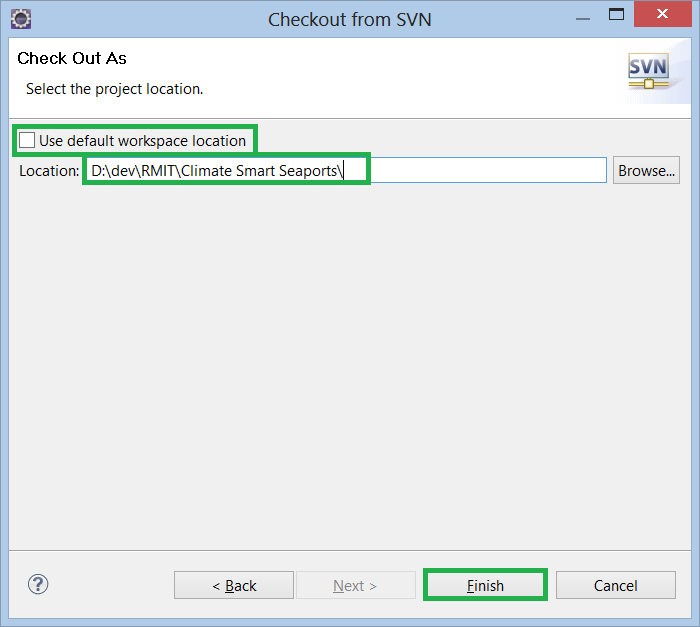
Wait while the project information are retrieved.



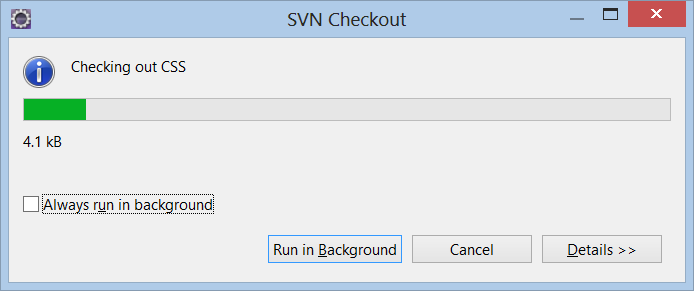
Select the option “Check out as a project in the workspace” and type a name for the new local project. To avoid any confusion, it is recommended to type the same project name as the remote project: “CSS”.

Make sure to check out the HEAD revision, fully recursive.

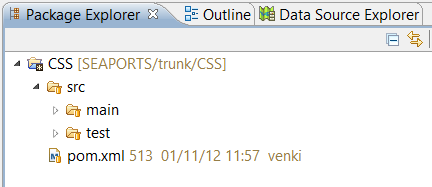
Click “Next”.



Uncheck the “Use default workspace location” checkbox, and browse for a location for the new local project. The process will create a subfolder for in the selected folder with the name of the project entered at the previous step.



Wait until the project is checked out. The project should appear in the package explorer View under Eclipse:

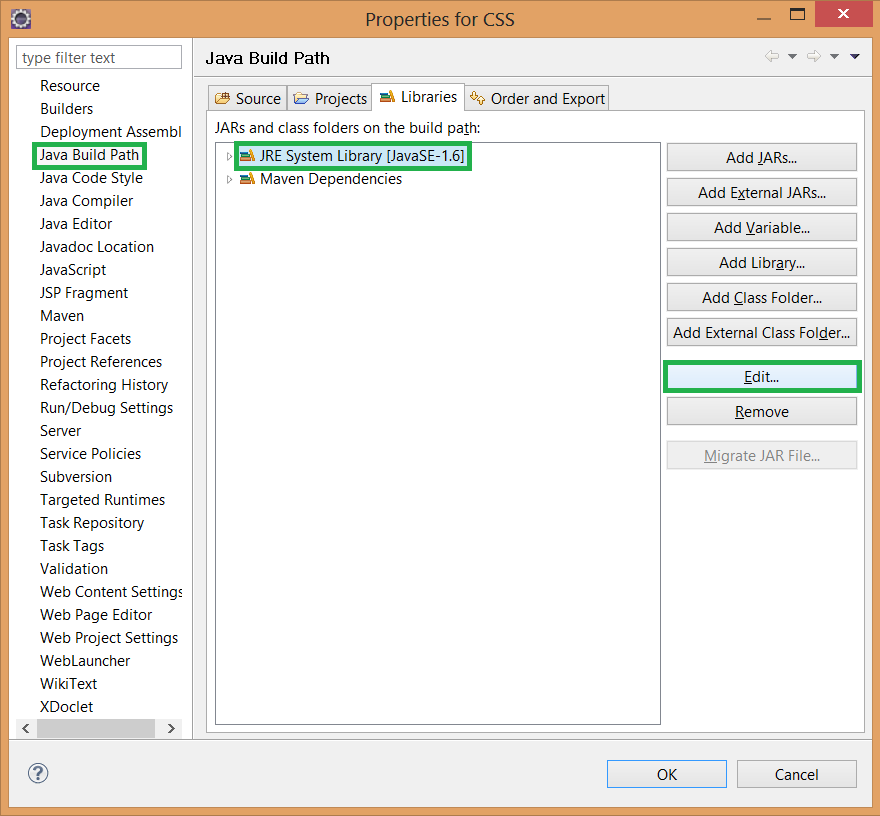


## Troubleshoot project issues

### Select the right jre System Library

To be able to compile and run, the project has to use JavaSE 1.6 from a Java Development Kit. If it uses the Java platform offered by a JRE instead of a JDK, some necessary features will be missing and make compilation fail. To configure the correct compilation settings, apply the following steps:

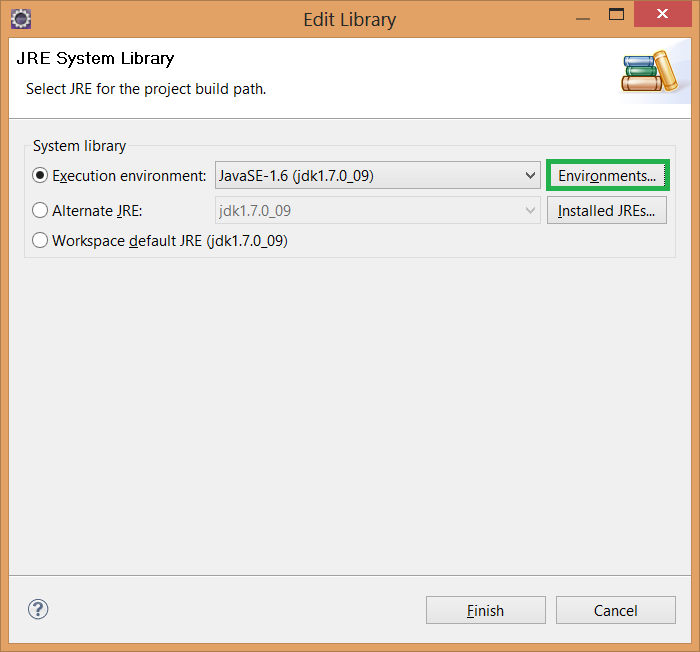
In Eclipse, right click on the project and select “Properties”.



In the project properties, select “Java Build Path from the list on the left.

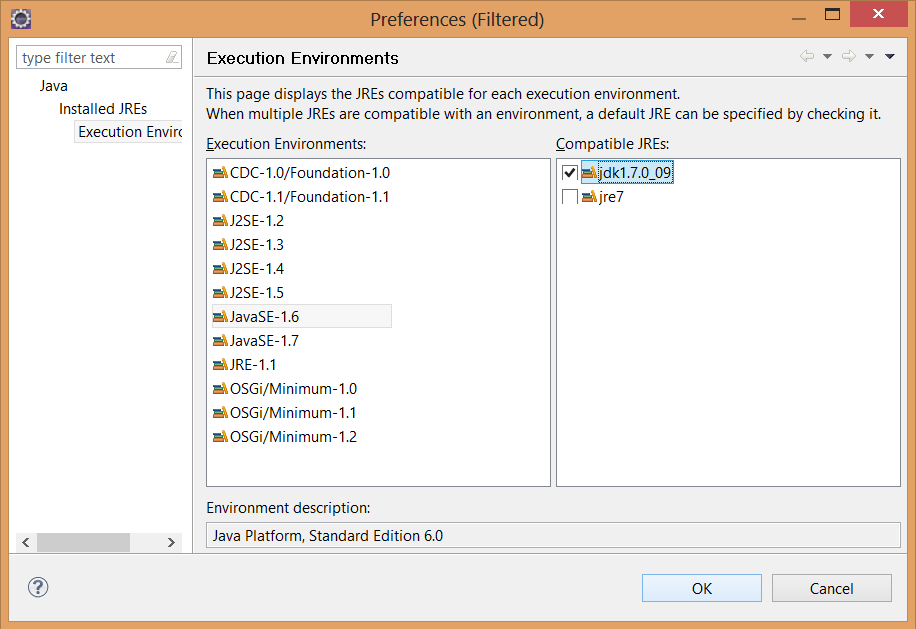
In the “Libraries” tab, select the “JRE System Library”

Click on “Edit…”.



In the Edit Library window, check that the execution environment in “JavaSE-1.6 (jdk…)”

If it’s not the case, click on “Environments…” to set it.



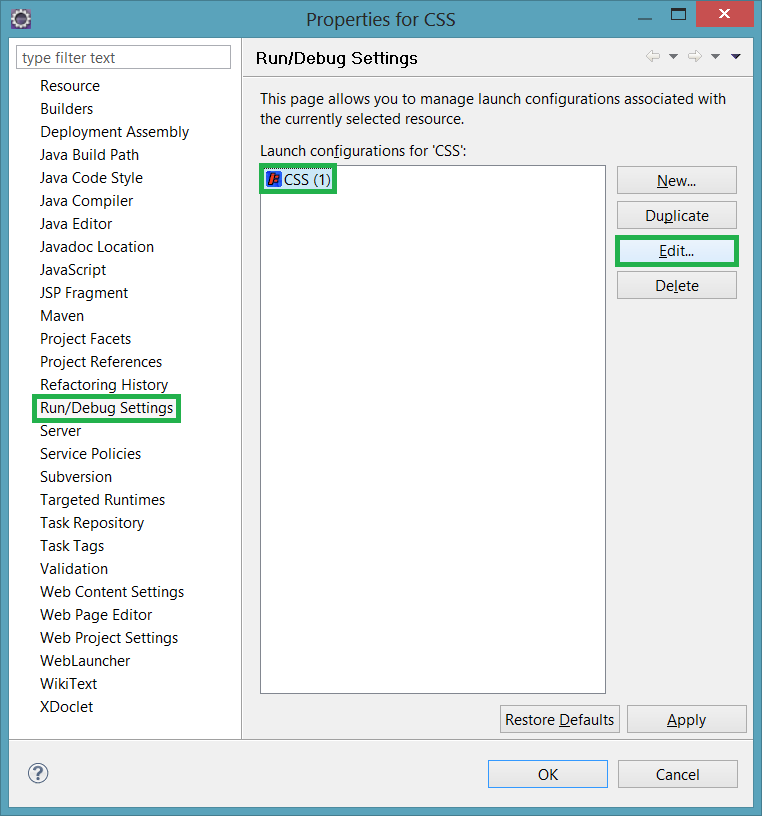
This opens the Execution environment preferences. Select JavaSE1.6 and check the “jdk” checkbox instead of the “jre”.

Click on “ok to validate all the settings.

### Select the right Run configuration

The project has to be run using Jetty 7.5

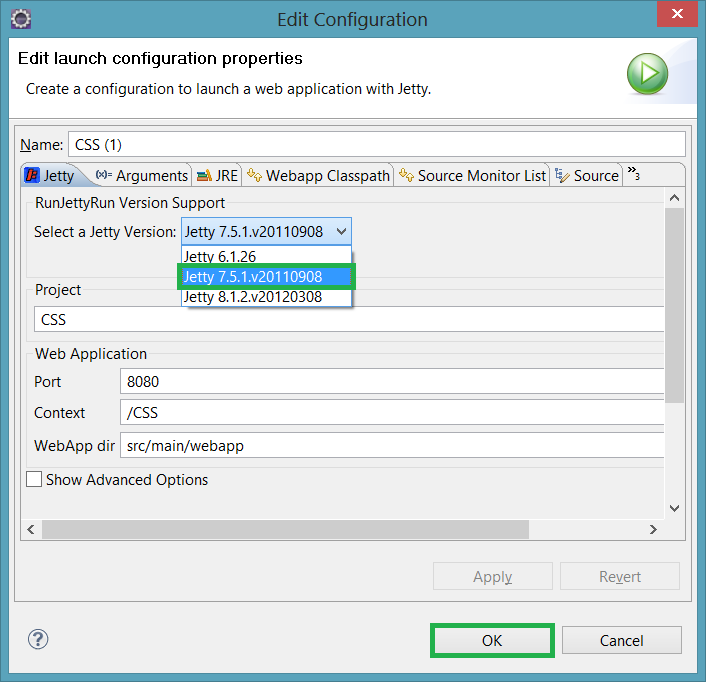
To set this, right click on the project in Eclipse, and select “Properties”.



Select the “Run/Debug Settings” in the list on the left.

Select the Jetty run configuration (Blue and Red icon, as shown in the above picture).

Click on “Edit…”



In the “Jetty” tab of this configuration window, select the version “Jetty 7.5.1”

Click “Ok” to save the settings.

### File upload issues at runtime

If an exception is raised when uploading a file this can be due to several issues:

1. The MySQL table field defined with the incorrect type of data: the automatic generation of SQL tables from the code thanks to Hibernate generates “blob” fields as “Tiny Blob” in MySQL as it should be “Long Blob”.

**Fix:** Open MySQL Workbench, connect to the MySQL database created for the project. Edit the “Files” table and change the type of data

1. The maximal allowed size limit in the project and the file is too big.

**Fix:** The maximal allowed file size can be changed in the following configuration file:

*“src/main/webapp/WEBINF/spring/app/servlet.context.xml”*.Modify the property “maxUploadSize” to a large number so that the line looks like the following:

*<beans:property name="maxUploadSize" value="100000000"/>*

1. You may also have to edit the MySQL configuration to let MySQL allow bigger packets. To do so, edit the “*max\_allowed\_packets*” variable in the “*my.ini*” file.

# Deployment under UNIX

## Deployment Pre-requisites

This section details the steps to install the all the prerequisites to the Climate Smart Seaports application deployment. The following instruction are valid for the UNIX distribution that have been used to deploy the application: **Red Hat Enterprise Linux Server release 6.3**

First, make sure you log in on the UNIX machine with a user having the necessary rights to install software on the machine.

### Java

Install the latest Java version (*jdk 1.7.0*).

Do to so, enter the following command in the terminal:

**$>yum install java**

Wait until the download and installation of Java is finished. You can then check that the installation was successful by typing:

**$>java -version**

The command should display the java version, starting with: *java version “1.7.0”*.

Set the JAVA\_HOME path variable, and add it to the PATH variable. For example:

**$>export JAVA\_HOME=/usr/java/jdk1.7.0  
  
$>export PATH=$JAVA\_HOME/bin:$PATH**

### Maven

Download the latest version of Maven (here 3.0.5). You should get the file: *apache-maven-3.0.5-bin.tar.gz*

Uncompress the Maven file into the desired folder (here */usr/local*) using the command:

**$>sudo tar xzf apache-maven-3.0.5-bin.tar.gz -C /usr/local**

Set the M2\_HOME path variable and add it to the PATH variable. For example:

**$>export M2\_HOME=/usr/local/maven  
$>export PATH=$M2\_HOME/bin:$PATH**

Configure Maven by creating a directory called “*.m2*” in your */home* directory. This is the Maven configuration folder. Copy the file settings.xml located in the “*conf*” directory of your Maven installation folder (here */usr/local/apache-maven-3.0.5/*):

**$> cd $HOME**

**$> mkdir .m2**

**$> cp /usr/local/apache-maven-3.0.5/conf/settings.xml $HOME/.m2/**

### Jetty

Move to the etc/yum.repos.d folder and add the Jetty repository:  
**$> cd /etc/yum.repos.d  
$> wget http://jpackage.org/jpackage.repo**

Install Jetty:

**$> sudo yum install jetty6**

Go to the Maven settings file *(/home/YOURUSERNAME/.m2/settings.xml*) and make sure the following line is added to in *pluginGroups*. If it isn’t there, add it.

extract of /home/YOURUSERNAME/.m2/settings.xml

<pluginGroups>

**<pluginGroup>org.eclipse.jetty</pluginGroup>**

<pluginGroups>

### MySQL

Install MySQL :

**$> yum install mysql mysql-server**

Start MySQL service (*mysqld*):

**$> service mysqld start**

Set the run level for the MySQL service (*mysqld*):

**$> chkconfig –level 235 mysqld on**

Run the secure installation of MySQL:

**$> /usr/bin/mysql\_secure\_installation**

Type the password when prompted and follow the steps of the installation until it’s finished.

## Installation

It is assumed that all the prerequisites are set up properly on the server. Connect to the server’s terminal using SSH.

If this is the first time, you can retrieve the project’s sources hosted on [Google code](http://climate-smart-seaports.googlecode.com) using SVN:

**$> svn checkout**

**http://climate-smart-seaports.googlecode.com/svn/trunk/ climate-smart-seaports-read-only**

Otherwise, retrieve the latest version or the sources. From a SVN repository, the command is:

**$>cd folder/where/the/sources/are/located**

**$>svn update**

Make sure that the application is properly configured (database credentials for hibernate, server name, port, application name, etc.). This is achieved by editing the application’s configuration files.

Please refer to the developer documentation for details about the application configuration.

### Database deployment

Run MySQL and load the SQL dump files:

**$mysql –u username –p**

Mysql prompts to enter password:

**$password**

(make sure the databases *seaports* and *seaports\_test* don’t exist already)

**mysql>source /path/to/the/file/seaports\_dump.sql**

**mysql>source /path/to/the/file/seaports\_test\_dump.sql**

(both *seaports\_dump.sql* and *seaports\_test\_dump.sql* should be located in the folder: *src/main/java/database* from the project folder).

**mysql>exit**

### Application Deployment

Clean:

**$>mvn clean**

Dependencies download, installation, running unit tests, copying compiled files to Web App:

**$>mvn install**

If the application was already running, kill it before deploying the Web application.

Web App deployment as a background process, and redirecting output to /dev/null:

**$>nohup mvn -Djetty.port=8080 jetty:run > /dev/null &**

The command to deploy the web application is as follows: *$mvn -Djetty.port=8080 jetty:run*, but we usually use the one above so that the process keeps running after the terminal is closed.

Try launching the URL:

***http://name.of.the.server:8080/***

The home page should be shown.