Eclipse and Maven Installation

# Prerequisites

In the following,

* <seuhome> refers to the directory the project environment has been extracted to, e.g. C:\Projekte\SEU\
* <soe-repository> refers to the directory the GIT repository containing the SOE source code has been cloned into, e.g. C:\Projekte\SOE\SOE\_Repository\

# Download and Extract the Setup File

Download the installation package file for the project environment and extract it locally (will be referred to <seuhome> in the following text). The folder now should contain the following folders.

* conf - Configuration files
* eclipse - Eclipse.
* java - Contains the JDK 6.
* software - Optional software (Contains SVN, squirrel-sql-3.3.0-install.jar & db2jars).
* tools - Maven 2 and 3.

**Note:** The configuration files are located in two places:

* <seuhome>\conf
* <soe-repository>\SOE-Build-Tools\src\main\resources

# Set Environment Variables

## Set the Environment Variables

Create the following system environment variables in Windows:

JAVA\_HOME=<seuhome>\java\jdk1.6.0\_22

M2\_HOME= <seuhome>\tools\apache-maven-3.0.4

## Set the Path

Now append the following to the PATH variable:

%JAVA\_HOME%\bin;%M2\_HOME%\bin;

# Setup Eclipse and the Workspace

## Set Default JVM for Eclipse

Open the file <seuhome>/Eclipse/eclipse/eclipse.ini. Replace the placeholder <seuhome> in this file with your SEU home or alternatively exchange the whole path to point to your local Java installation.

## Setup the Proper Workspace in Eclipse

Open Eclipse. Go to File → Switch Workspace and select the directory in which to set up the Eclipse workspace, e.g. C:\Projekte\SOE\.

## Configure Maven in Eclipse

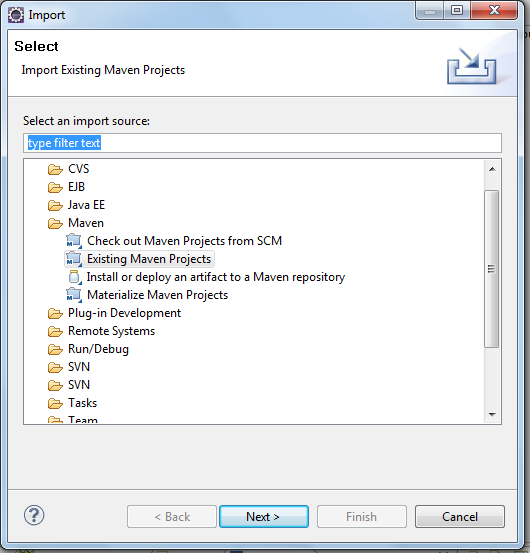
Copy the file <soehome>\conf\Maven-Settings\settings.xml to %USERPROFILE%\.m2. The file contains the initial configuration for Maven.

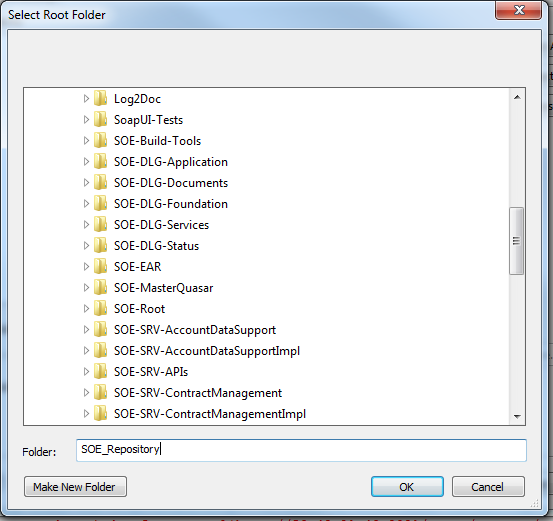
In Eclipse choose Window → Preferences → Maven → Installations. Click on Add and select the directory M2\_HOME (<seuhome>\tools\apache-maven-3.0.4\). Open Maven → User Settings. In the first text box you can see the path to the settings file settings.xml that is located in the %USERPROFILE%\.m2 folder.

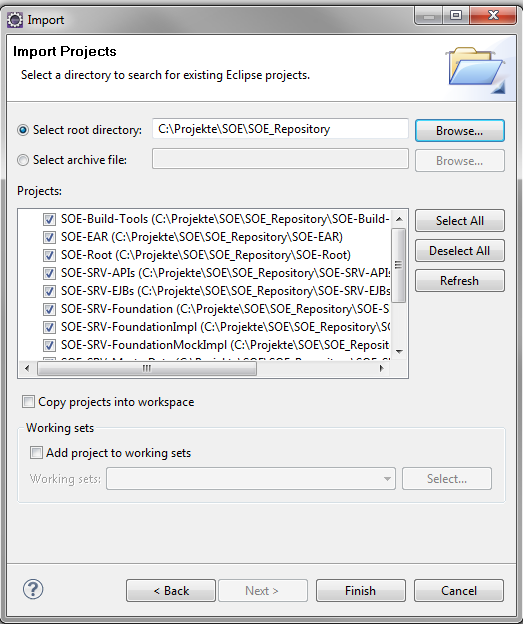
**Note:** Here %USERPROFILE% means the folder which can be found under C:\Users\.

## Add Projects to Workspace

In the menu bar choose File → Import and choose Maven → Existing Maven Projects. Select the location that you checked out the SOE code from the GIT repository to and press Finish.

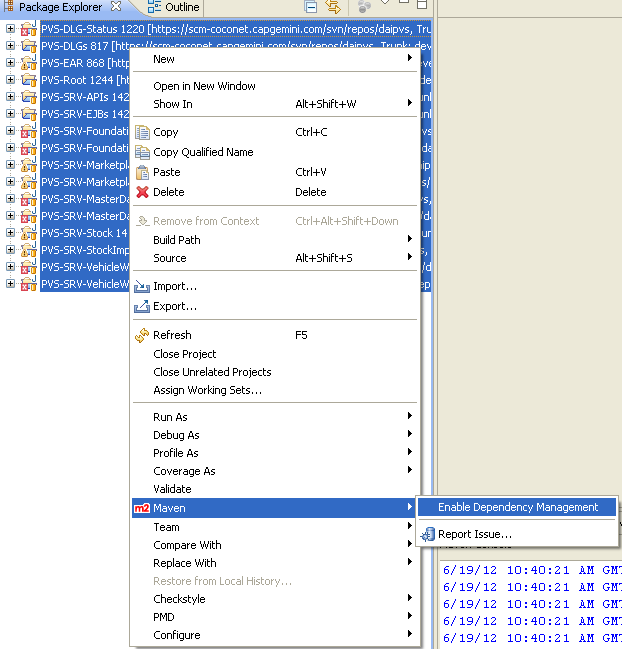






## Enable Maven Dependency Management

Change the perspective to Java EE and select all projects. Right-click on the projects and choose Maven → Enable Workspace Resolution and Maven → Enable Dependency Management (if necessary).



## Configure Checkstyle, PMD and FindBugs

### Configure Checkstyle

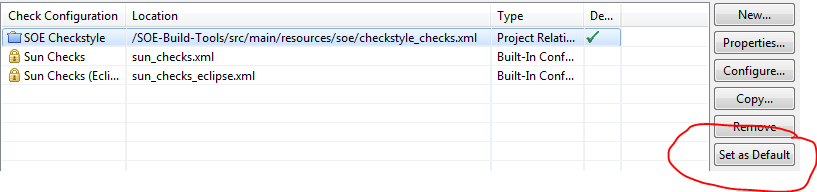
Copy the Checkstyle file <soe-repository>/SOE-Build-Tools/src/main/resources/soe/checkstyle\_checks.xml to a location outside your local repository.

In the menu bar choose Window → Preferences → Checkstyle and click on New.

Choose the type External Configuration File, set the name to SOE Checkstyle Settings and import the file you just copied.

Exclude the test files and the generated files from the analysis by clicking on Configure. Select Filters → Suppression Filter, click on Add and import the exclusion file <soe-repository>/SOE-Build-Tools/src/main/resources/soe/checkstyle\_suppressions.xml.

Make the imported Checkstyle configuration default by clicking Set as Default.



Furthermore you can deactivate the entry Todo Comment in the configuration dialog of Window → Preferences → Checkstyle → SOE Checkstyle Settingsin the tab Miscellaneous*.*

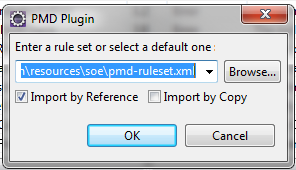
Analyze the code by right-clicking the Eclipse project and selecting Checkstyle → Activate Checkstyle.

### PMD

In the menu bar choose Window → Preferences → PMD → Rules Configuration.

Click on Clear all.

Click on Import rule set and select the file with the PMD rules in <soe-repository>/SOE-Build-Tools/src/main/resources/soe/pmd-ruleset.xml.



Import the rule set by reference.

Click on Add Exclude Pattern and exclude the following locations:

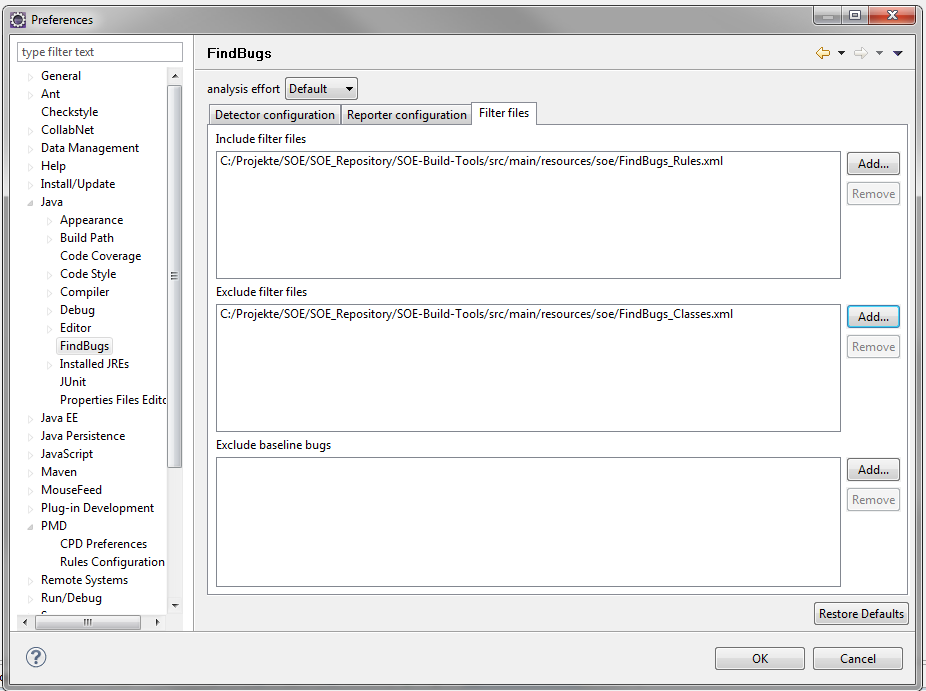
* .\*/src/test/.\*
* .\*/gen/main/java/.\*

Analyze the code by right-clicking on the Eclipse project and selecting PMD → Check Code with PMD.

To get Eclipse to not flag the @SuppressWarnings("PMD") annotation, look under the menu headings Java → Compiler → Errors/Warnings → Annotations → Unhandled Token in '@SuppressWarnings' and set it to ignore.

### FindBugs

In the menu bar choose Window → Preferences → Java → FindBugs → Filter files. Import the Find Bugs rule set by adding the file <soe-repository>/SOE-Build-Tools/src/main/resources/soe/FindBugs\_Rules.xml to the Include filter files and by adding the file <soe-repository>/SOE-Build-Tools/src/main/resources/soe/FindBugs\_Classes.xml to the Exclude filter files.



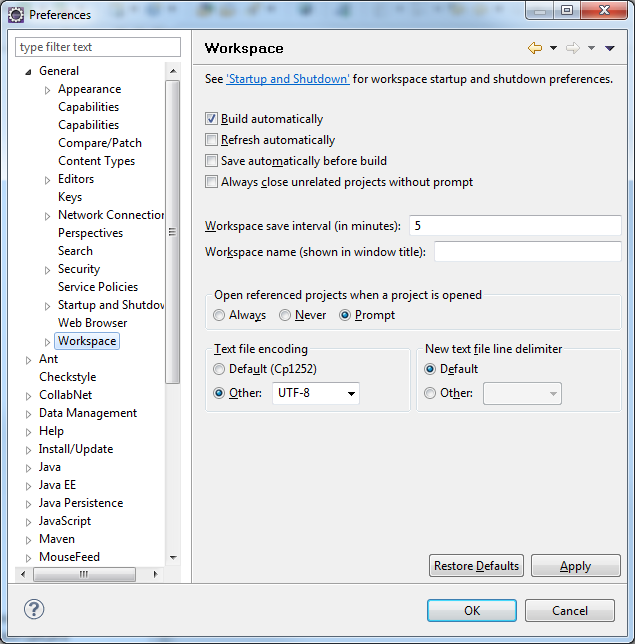
## Set UTF-8 Encoding

In the menu bar choose Window → Preferences → General → Workspace and set UTF-8 as default encoding.

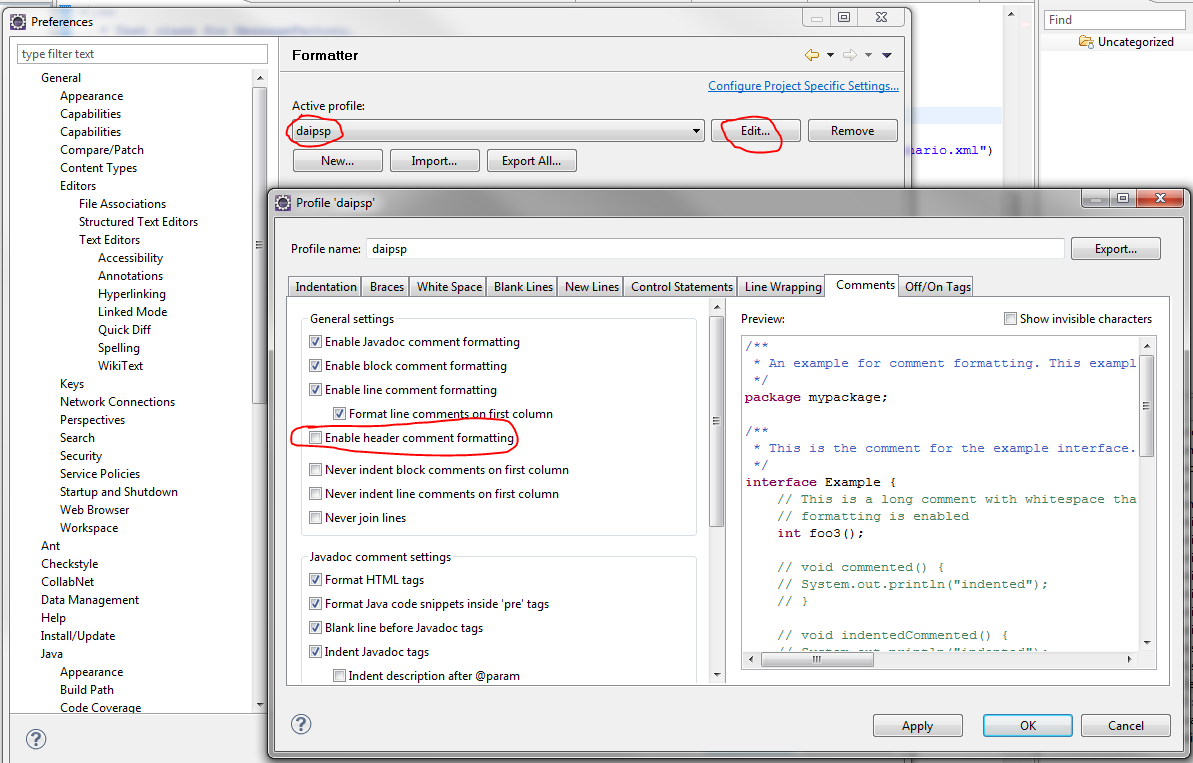
## Additional Eclipse Settings

### Code Formatting

In the menu bar choose Window → Preferences → Java → Code Style → Formatter → Import and select the file <seuhome>/conf/formatter.xml.



Click on Edit → Comments and uncheck Enable header comment formatting to disable the predefined header comment formatting.



### Code Clean Up

In the menu bar choose Window → Preferences → Java → Code Style → Clean up → Import and select the file <seuhome>/conf/cleanup.xml.

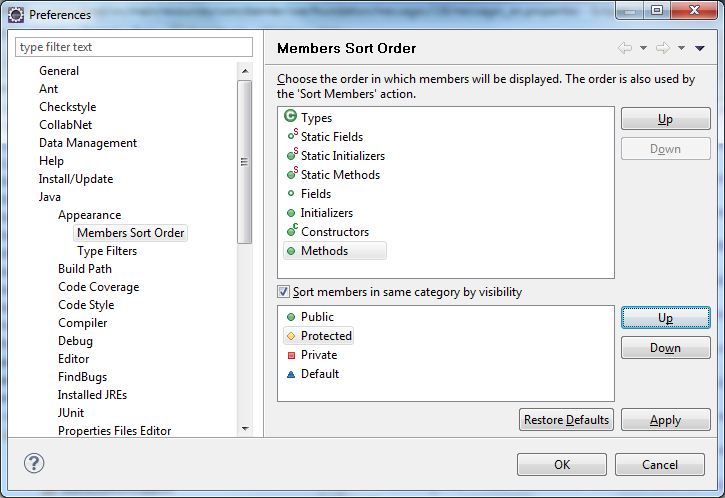
Instead of manual cleaning up the code, the Auto Save functionality can be configured. In the menu bar choose Window → Preferences → Java → Editor → Save Actions. Check Perform the selected actions on save and check the actions to automatically perform when saving a file, e.g. Format source code and Organize Imports.

### Code Templates

In the menu bar choose Window → Preferences → Java → Code Style → Code Templates → Import and select the file <seuhome>/conf/codetemplates.xml.

### Setting the Sort Order of methods (and fields) in Eclipse

In the menu bar choose Window → Preferences → Java → Appearance → Members Sort Order.

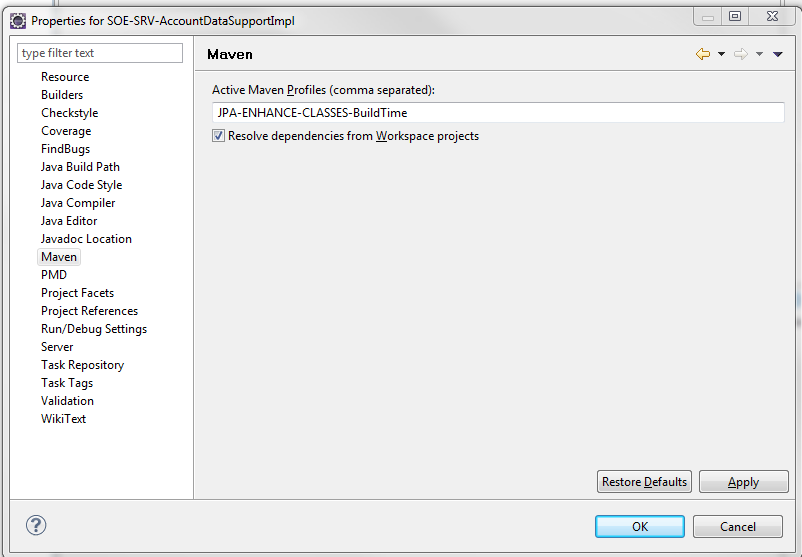


* Tick the box Sort members in same category by visibility
* Change the default order: Pull Protected one up, so that the result reads Public > Protected > Private > Default as shown above.

**Hint:** This order applies when reformatting a class in the editor (e.g. using Ctrl + Shift + F). If you mark a section or several lines prior to applying the eclipse formatter, only the marked code will be formatted. That is sometimes quite useful.

## Enable JPA enhancement on Maven build in Eclipse

For SOE-COM-TechBase and each \*Impl project, right-click the project, go to Properties → Maven and enter JPA-ENHANCE-CLASSES-BuildTime as Maven profile.



## Build the Project

1. Right-click the project SOE-Build-Tools and choose Run As → Maven install.
2. Right-click the project SOE-Root and choose Maven → Update Project Configuration.
3. Right-click the project SOE-Root and choose Run As → Maven build. Select the launch configuration package-no-tests: package.

The build should work without any problems.

# Initialize the Local Database

You need to run the following SQL scripts on your local machine after replacing the path (highlighted in bold) e.g. using Eclipse or SQuirreL:

-- bufferpool for data of normal tables

CREATE BUFFERPOOL BP1\_8  SIZE 5000 PAGESIZE 8192;

-- bufferpool for indexes of normal tables

CREATE BUFFERPOOL BP2\_8  SIZE 5000 PAGESIZE 8192;

-- tablespace for data of normal tables

CREATE LARGE TABLESPACE VMGAD18 PAGESIZE 8192

MANAGED BY DATABASE

USING (FILE 'C:\Projekte\SOE\VMGAD18.dbf' 30000)

EXTENTSIZE 32

        PREFETCHSIZE AUTOMATIC

AUTORESIZE YES

BUFFERPOOL BP1\_8

MAXSIZE NONE

NO FILE SYSTEM CACHING

DROPPED TABLE RECOVERY ON;

-- tablespace for indexes of normal tables

CREATE LARGE TABLESPACE VMGAI18 PAGESIZE 8192

MANAGED BY DATABASE

USING (FILE 'C:\Projekte\SOE\VMGAI18.dbf' 10000)

EXTENTSIZE 32

PREFETCHSIZE AUTOMATIC

AUTORESIZE YES

BUFFERPOOL BP2\_8

MAXSIZE NONE

NO FILE SYSTEM CACHING

DROPPED TABLE RECOVERY ON;

-- tablespace for LOBs of normal tables

CREATE LARGE TABLESPACE VMGAL18 PAGESIZE 8192

MANAGED BY DATABASE

USING (FILE 'C:\Projekte\SOE\VMGAL18.dbf' 30000)

EXTENTSIZE 32

PREFETCHSIZE AUTOMATIC

AUTORESIZE YES

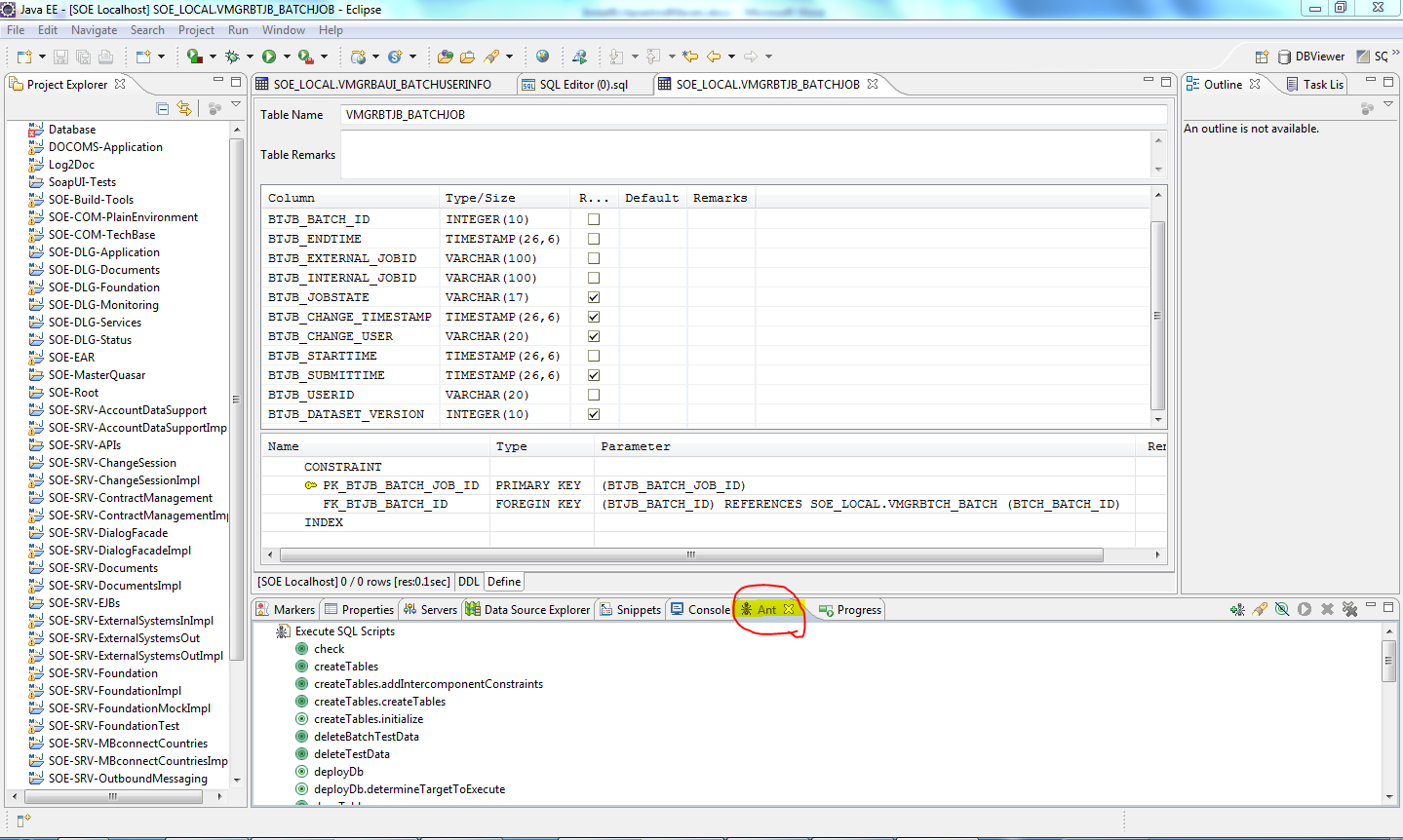
BUFFERPOOL BP1\_8

MAXSIZE NONE

NO FILE SYSTEM CACHING

DROPPED TABLE RECOVERY ON;

Go to the project Database and open the file executeSqlScripts.xml. In the outline of the file (Alt + Shift + Q, O), right-click on resetTablesAndImportBaseAndTestData and choose Run As → Ant Build or go to the Ant tab and double click on the corresponding SQL Script.



## Human Readable SQL Errors

To get human readable SQL error messages, add the parameter retrieveMessagesFromServerOnGetMessage=true within SQuirreL to the properties of your SQL Connection profile.

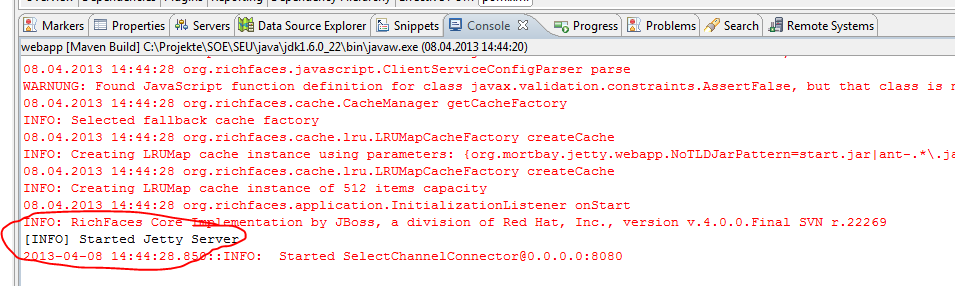
# Setup Jetty for Local GUI Testing

## RunJettyRun Plugin for Eclipse

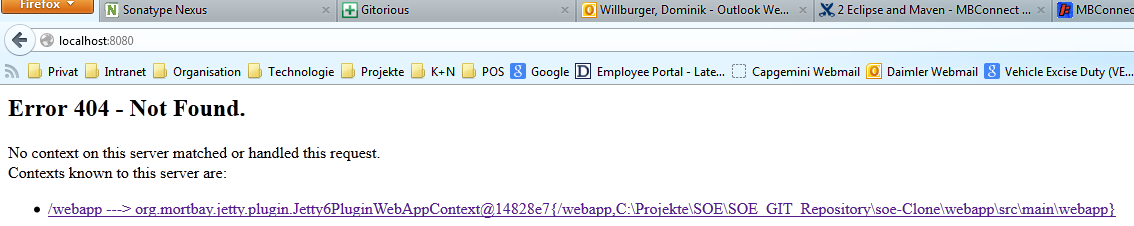
Install the RunJettyRun [plugin](http://run-jetty-run.googlecode.com/svn/trunk/updatesite).

## Run the Local Web Application

The local web application is executed by right-clicking the project SOE-WAR and choosing Run-As → Run Jetty. The console should show something like this:



The application should now be available on any local browser on the URL http://localhost:8080/soe-webapp. This page should display a link which takes you to the application.



# Installing the ResourceBundle Editor

The Eclipse plugin ResourceBundle Editorcan be used in order to easily maintain resource bundles.

In the menu bar choose Help → Install New Software and add the repository <http://www.nightlabs.de/updatesites/development/> to the list of available sites. Select the ResourceBundle Editor Plug-in and click on Install.