0.[Overview 3](#_Toc20397)

[Notes: 5](#_Toc21496)

[Account Creation: 5](#_Toc28280)

[CVS Account Creation 6](#_Toc21680)

1.[Local Environment Configuration Overview 6](#_Toc22000)

1.1[MySQL 6](#_Toc17095)

[Installation: 6](#_Toc20074)

[MySQL Configuration File: my.cnf  7](#_Toc2364)

[MySQL database for Real Time FM 7](#_Toc775)

[1. Create 'srs' database with full privilege user. 7](#_Toc3218)

[2. Create initial 'attachments' table within 'srs' database 8](#_Toc27166)

[3. Import main data into 'srs' database 8](#_Toc1566)

[4. Verify 'srs' database 8](#_Toc27302)

[MySQL database for Android Messaging Gateway (AMG) 8](#_Toc20519)

[1. Create 'amg' database with full privilege user. 8](#_Toc28742)

[Import main data into 'amg' database 9](#_Toc16433)

1.2[JBoss 10](#_Toc6409)

[Installation: 10](#_Toc26726)

[Environmental Variables 10](#_Toc4498)

[Files Requiring Review/Updates: 10](#_Toc8567)

[Run/Verify JBoss 10](#_Toc7166)

1.3[Tomcat 14](#_Toc2165)

[Installation: 14](#_Toc24286)

1.4[Android SDK 15](#_Toc14440)

[Installation: 15](#_Toc17641)

2.[Eclipse 15](#_Toc1461)

[Configure CVS 16](#_Toc29066)

[RealTime FM and Related Project Check Out 20](#_Toc19468)

[Eclipse Coding Standard 21](#_Toc32244)

[Code Style 21](#_Toc14822)

[Code Save Action 23](#_Toc3906)

[Configure JBoss Server 26](#_Toc14597)

[Configure Apache Tomcat 27](#_Toc11748)

[Configure Dependencies 28](#_Toc27074)

[Configuring ANT  30](#_Toc21482)

[Configure Android Development Tools Plugin 39](#_Toc13464)

[Configure Project from Eclipse with ADT 39](#_Toc4643)

[Configure Android Virtual Device (AVD) 42](#_Toc10159)

3.[RealTime FM Configuration 44](#_Toc31357)

[Add New User to RealTime FM 45](#_Toc16049)

[Add New Ticket to RealTime FM 45](#_Toc5997)

[Appendices 46](#_Toc19359)

[Troubleshooting 46](#_Toc10376)

[New Tasks generated in RealTime are failing to display on Taskman application. 46](#_Toc18240)

[Clearing/Resetting Taskman Android Application 46](#_Toc9461)

[Resend Taskman user credentials 46](#_Toc4013)

[RealTime FM Reporter Service 46](#_Toc28590)

4.[Database upgrade utility 52](#_Toc11293)

[Overview 52](#_Toc27309)

[Schema management 53](#_Toc30673)

[Migration script 53](#_Toc18584)

[Database upgrade utility 53](#_Toc13728)

[Project build path](#_Toc13728) [5](#_Toc18584)4

[Overview 67](#_Toc14498)

[Schema management 67](#_Toc22714)

[Migration script 67](#_Toc9928)

[Database upgrade utility 68](#_Toc29841)

Client:       ARAMARK, Ireland

Location:     Cork, Ireland

Author:        Ger O'Sullivan

Date:           2012-08-02

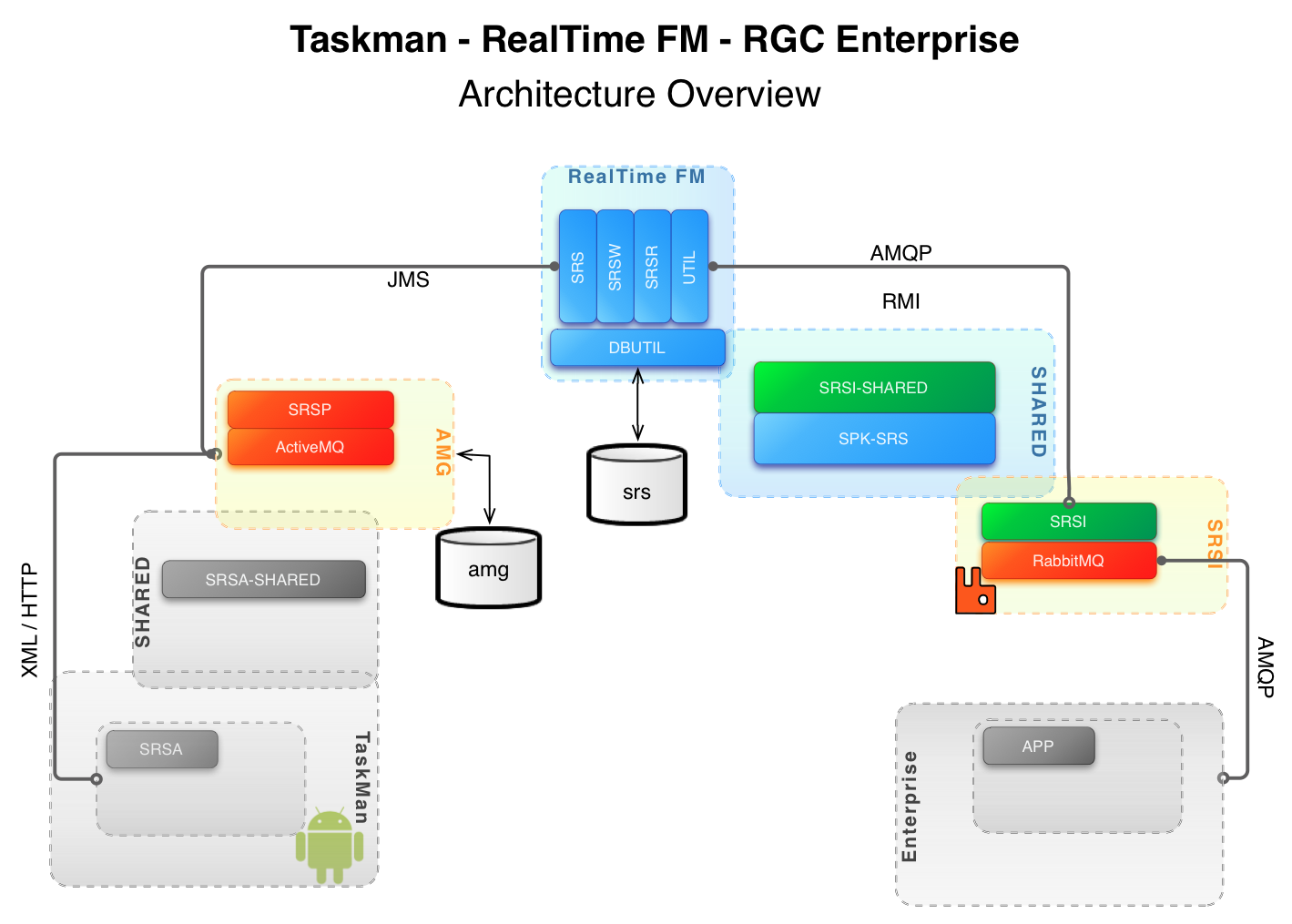
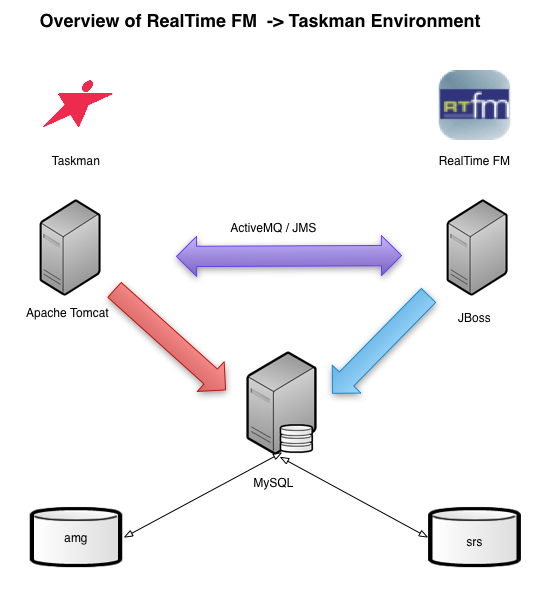
Edited By: Johnny Wu

Last Edited: 2014-07-25

Overview

The following is documentation outlining the steps followed regarding the configuration environments for the following:

* [MySQL](http://en.wikipedia.org/wiki/MySQL) (Optimal version \*\* 5.5.1 \*\*)- open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.
* [Concurrent Versions System (CVS)](http://en.wikipedia.org/wiki/Concurrent_Versions_System) - is a client-server free software revision control system in the field of software development.
* [Eclipse](http://en.wikipedia.org/wiki/Eclipse.org) (3.7) / [MyEclipse](http://en.wikipedia.org/wiki/MyEclipse) (10) - a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system.
* Eclipse Plug-ins - See section titled Eclipse Plugins.
* [JBoss](http://en.wikipedia.org/wiki/JBoss) Application Server - an application server that implements J2EE.
* [Apache Ant](http://en.wikipedia.org/wiki/Apache_Ant) - software tool for automating software build processes.
* [ActiveMQ](http://en.wikipedia.org/wiki/Apache_ActiveMQ) - an open source messaging broker which fully implements the Java Messaging Service 1.1 (JMS).
* [Android SDK](http://en.wikipedia.org/wiki/Android_software_development) - Applications are usually developed in the [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) programming language using the Android [Software Development Kit](http://en.wikipedia.org/wiki/Software_Development_Kit)
* [Redmine](http://en.wikipedia.org/wiki/Redmine)  - is a free and open source, web-based project management and bug-tracking tool. It includes a calendar and Gantt charts to aid visual representation of projects and their deadlines.



Notes:

Currently, there is a server located on the first floor of the Cork office that houses the following applications/systems.

Server:

* alm.aramark.eu

Apps:

* CVS.
* Redmine.

Account Creation:

The following applications require the creation of a new account.

* CVS
* Redmine

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THIS IS FOR FUTURE CODE REPOSITORY ADMIN USE ONLY \*\*\*/

CVS Account Creation

Setting up a new CVS user

--------------------------------------------------------------------------------

Host: [alm.aramark.eu](http://alm.aramark.eu/)

// Add new user

$ sudo useradd -d /home/<username> -m <username>

// Change user password

sudo passwd <username>

// Modify user to add user and cvs group

sudo usermod -a -G users <username>

sudo usermod -a -G cvs <username>

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End admin section \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

Local Environment Configuration Overview

* Local Working (Project) Copies - checked out from CVS
* Savant Repository
* MySQL
* Eclipse
* JBoss server
* Apache Tomcat Server
* ActiveMQ
* <AWS\_CVS\_BUILD\_FILES> - a bundled set of files that constitute a current development build environment. This file bundle is currently residing on the [alm.aramark.eu](https://alm.aramark.eu) server under the following directory:
  + /home/shared/rtfm/workspace/current

**Note: Download either putty or Winscp and logon w/ the short id (not the one w/ the domain extension)**

MySQL

Installation:

* Accept all default values

MySQL Configuration File: my.cnf

* Locate and edit the MySQL 'my.ini'.
* Find and locate the [mysqld] section
* Under the section [mysqld], add a new line 'lower\_case\_table\_names=1'
* Save 'my.cnf'
* From the command line, restart the MySQL instance.
* Open the **MySQL Workbench**
* Double click on the instance under **Server Administration**
* Click on Startup/Shutdown
* Click on the Stop Server button
* Click on the Start Server button
* Make sure the status (in green) says **running**.

MySQL database for Real Time FM

1. Create 'srs' database with full privilege user.

* Log into MySQL as 'root'
* Create a database titled 'srs'.
* Double click the Local instance
* In the query window, type *create database srs;*
* Click the on refresh button next to the word SCHEMAS on the left-hand pane.
* Create a new user 'srsadmin' with a default password 'admin123'. This will create a new user with full privileges for database 'srs'.

* + - Open MySQL Command 5.5 Command Line Client
    - Type the following:
* create user ‘srsadmin’@’*instance name*’ IDENTIFIED BY 'admin123';
* grant all privileges on srs TO 'sysadmin'@'*instance name*' WITH GRANT OPTION;
* create user ‘srsadmin’@’*%*’ IDENTIFIED BY 'admin123';
* grant all privileges on srs.\* TO 'sysadmin'@'%' WITH GRANT OPTION;

Each of the above statements should produce this return message:

*Query OK, 0 rows affected (0.00 sec)*

* Verify 'sysadmin' user account creation by logging into MySQL –
  1. open a cmd prompt
  2. cd to the “MySQL Server 5.5\bin” directory
  3. mysql -u**srsadmin** -p**admin123** -D **srs**

2. Create initial 'attachments' table within 'srs' database

Note:  Since the main MySQL data backup/dump excludes attachments, the attachments' table needs to be manually added to the 'srs' database.

* Navigate to **/home/shared/rtfm/workspace/builds/4.3.0/mysql/srs** from the ssh

session to alm.aramark.eu

* Download empty-attachments.sql
* Import data into newly created 'srs' database using the following terminal command

mysql -u**root** -D **srs** < **empty-attachments.sql**

3. Import main data into 'srs' database

* Extract a copy of the RealTimeFM MySQL data (~500MB) 'aws.sql' from the <AWS\_CVS\_BUILD\_FILES>
* Import data into newly created 'srs' database using the following terminal command

mysql -u**root** -D **srs** < *[[****srs\_sql\_dump\_file\_name.sql]]***

where *[[****srs\_sql\_dump\_file\_name.sql]]*** is the file name retrieved from the shared folder on **alm.aramark.eu**.

4. Verify 'srs' database

Verify data and content are successfully imported.

* Execute the following SQL Command to retrieve a count of the number of tables within the 'srs' database:

mysql> show tables;

* As of 2012-10-1, the 'srs' database contains the following table count.

+------------------+

| number\_of\_tables |

+------------------+

|               96 |

+------------------+

MySQL database for Android Messaging Gateway (AMG)

1. Create 'amg' database with full privilege user.

Log into MySQL as 'root'

Create a database titled 'amg'.

* Enter the below statements to create a new user 'amg' with a default password 'amg’. This will create a new user with full privileges for database 'amg'.

        create user amg@CelebiMySQL IDENTIFIED BY 'amg';

grant all privileges on amg.\* TO amg@CelebiMySQL WITH GRANT OPTION;

* Verify 'amg' user account creation by logging into MySQL -

          mysql -u**amg** -p**amg** -D **amg**

1. Import main data into 'amg' database

mysql -u**root** -D **amg** < *[[****amg\_sql\_dump\_file\_name.sql]]***

where *[[****amg\_sql\_dump\_file\_name.sql]]*** is the file name retrieved from the shared folder on **alm.aramark.eu**.

* Open a cmd prompt
* cd to cd to the “MySQL Server 5.5\bin” directory
* mysql -u**amg** -p**amg** -D **amg**
* Execute the following SQL Command to retrieve a count of the number of tables within the 'amg' database'

mysql> show tables;

* As of 2012-10-1, the 'amg' database contains the following table count.

+------------------+

| number\_of\_tables |

+------------------+

|               13 |

+------------------+

JBoss

* *denotes Windows specific instruction*

*\*\* Install the latest version of Java BEFORE continuing to the below section*

Installation:

The current build uses JBoss 4.0.4, which can be downloaded directly <AWS\_CVS\_BUILD\_FILES> or via <http://www.jboss.org/jbossas/downloads/> (??)

Extract jboss.tar.gz and install it into a directory of your choosing.

* Goto <http://sourceforge.net/projects/jboss/files/JBoss/JBoss-4.0.4.GA/>
* Download [jboss-4.0.4.GA.zip](http://sourceforge.net/projects/jboss/files/JBoss/JBoss-4.0.4.GA/jboss-4.0.4.GA.zip/download" \o "Click to download jboss-4.0.4.GA.zip)
* Extract the content of the zip file to c:\program files (x86) or whatever your Program Files folder is located

Environmental Variables

Add the following environmental variables to the your system PATH.

$JBOSS\_HOME   =     <jboss\_install>

$JAVA\_HOME     =      <java\_sdk\_install>

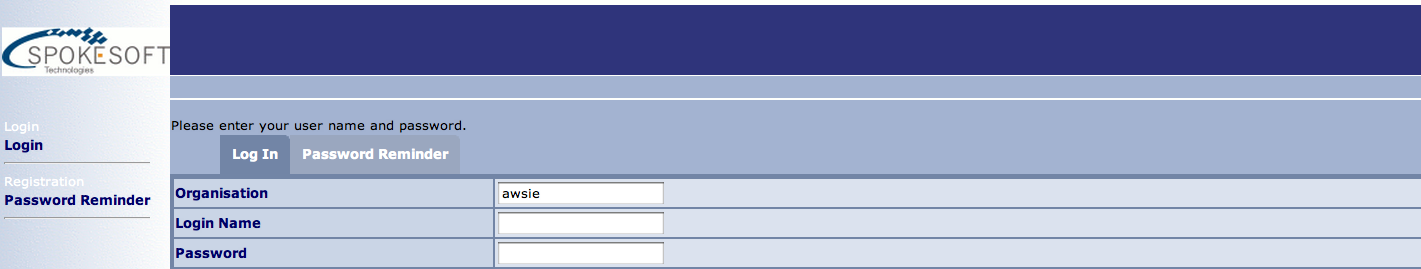
Files Requiring Review/Updates:

* + Copy all missing files from the <AWS\_CVS\_BUILD\_FILES> to <jboss\_install>/server/default/deploy directory

Run/Verify JBoss

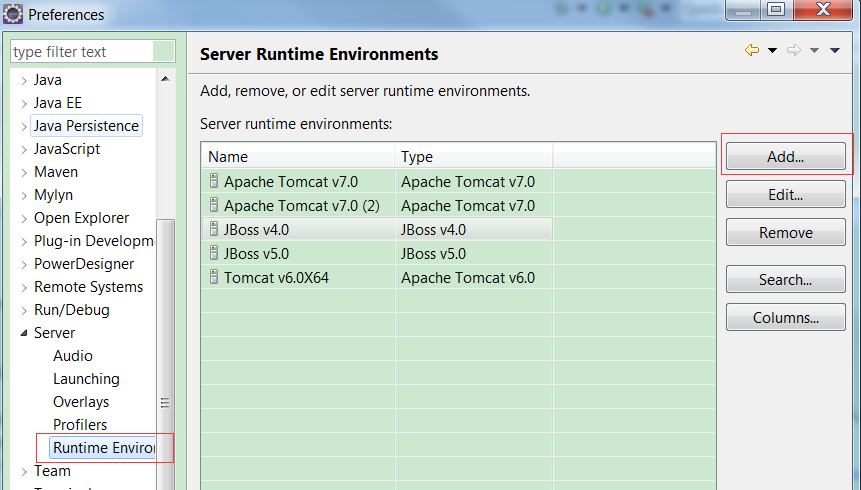
* Extract a copy of the spk-srs.ear file from <AWS\_CVS\_BUILD\_FILES>.
* Place the .ear file in the following location [<jboss\_install>/server/default/deploy/spk-srs.ear](http://jboss.org/4.0.4/server/default/deploy/spk-srs.ear)
* From the terminal, start JBoss by launching 'run' from within the <jboss\_install>/bin directory.\
* Ensure that the JBoss server is binding to port specified in server.xml. Troubleshoot any exceptions before proceeding to the next step.
* Launch url [http://localhost](http://localhost/):<port\_binding>.RealTimeFM login page should now display.
* Login using following admin account information (as defined in bundled MySQL database):
  + User Name:  admin
  + Password:    v3ct0r

If jboss cann’t connect mysql then copy ‘mysql-ds.xml’ to ‘jboss\_home\server\default\deploy\’ and ‘mysql-connector-java-5.1.23.jar or other edition’ to ‘jboss\_home\server\default\lib\’

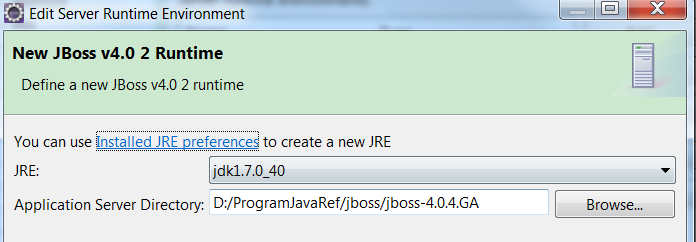
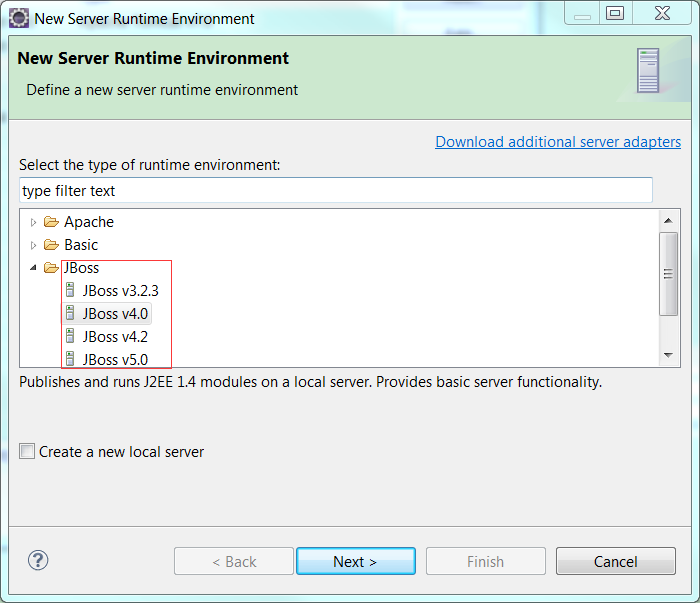


Add Jboss to Eclipse:

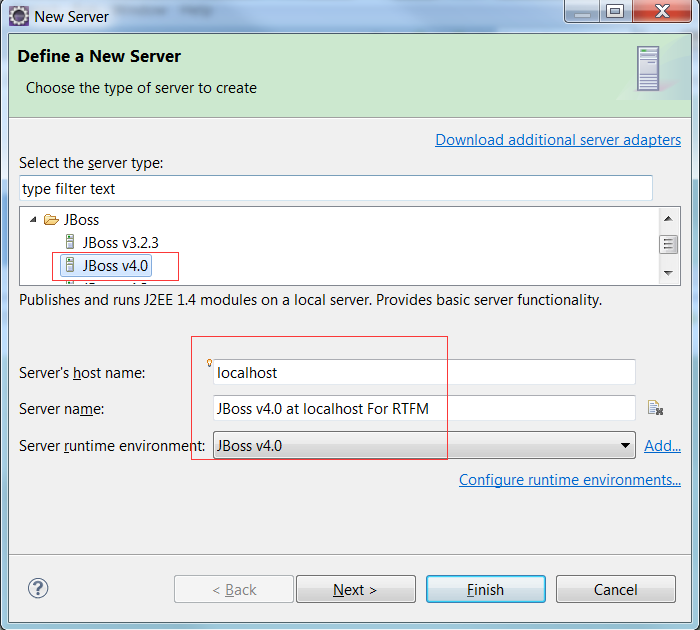
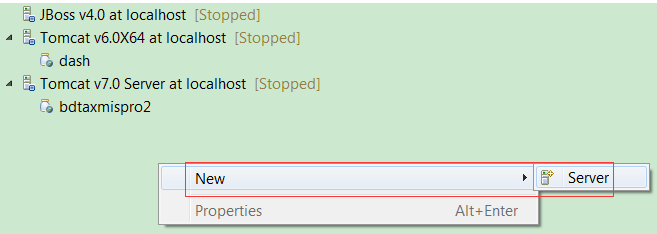
1. Open Eclipse Preferences and select ‘Server->Runtime Enviroement’. Click Add...



1. Select jboss, jre edition and jboss root directory.



1. Add new server to ‘Eclipse Servers’. Right click mouse button in the ‘Servers Tab’ blank space and select ‘New ->Server’. Then set the server port and add project ‘srs’ to ‘Configured’. Finished with **no** ‘Available resource’ added to ‘Configured’(Because the ant will send build package to your jboss runtime directory).



Tomcat

* *denotes Windows specific instruction*

Installation:

The current build uses Apache Tomcat 6.0.18, which can be downloaded directly <AWS\_CVS\_BUILD\_FILES> or via <http://tomcat.apache.org/download-60.cgi> (??)

Extract tomcat.tar.gz and install it into a directory of your choosing.

* Extract the zip file to the Program Files folder

The following files may require a review or update in order to fix workspace specific property settings (i.e. username, password, skins, binding):

<tomcat\_install>/conf/server.xml

* Property <connector port="8081">
* Property <connector keystoreFile="(path/to/ARAMARK-KEYSTORE-FILE)"/>

<tomcat\_install>/conf/tomcat-users.xml

* Property <tomcat-users>

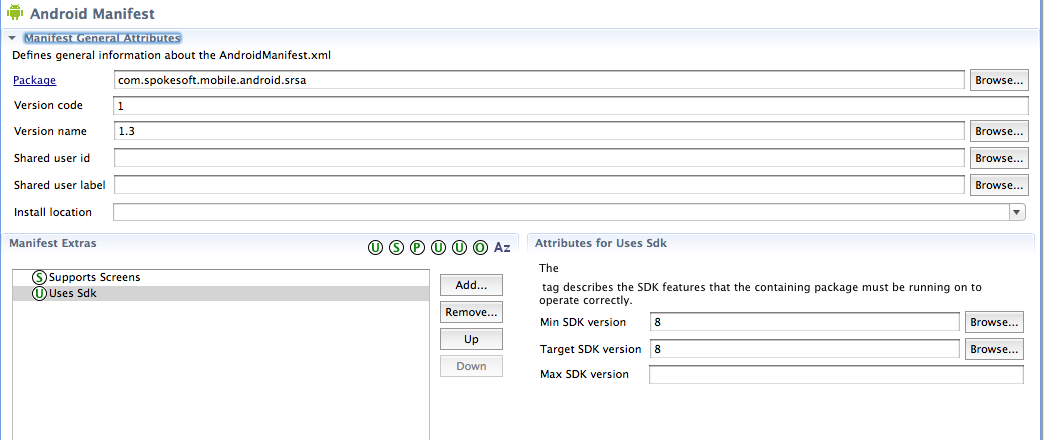
Android SDK

Installation:

The current build uses Android XX, which can be downloaded directly <AWS\_CVS\_BUILD\_FILES> or via <http://developer.android.com/sdk/index.html>

Extract android.tar.gz and install it into a directory of your choosing.

The current build uses SDK Version 8 which can be configured through the current SDK download.



Eclipse (*for Juno 4.2.0 for Java EE Developer*)

~~The current build uses Eclipse 3.7 for J2EE (codename Indigo).~~

~~Eclipse Plugin~~

Use the following screenshot from the Eclipse Software Center dialog screen to determine the plugins and the versions to be installed.

The main plugins required are the following:

1. ~~Eclipse CVS Client (default plugin with 3.7)~~
2. ~~Eclipse CVS Client Resources (default plugin with 3.7)~~
3. Android Development Toolkit

Configure CVS

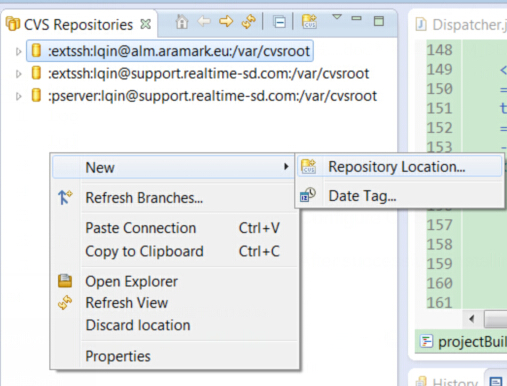
Neo (Juniper Network Connects, SSL VPN)

Once the Neo vpn connection has been established, the developers will be able to access the RealTime /FM environments as well as CVS over SSH.

To establish the vpn connection, navigate to <https://neo.aramark.eu/edcadmins>

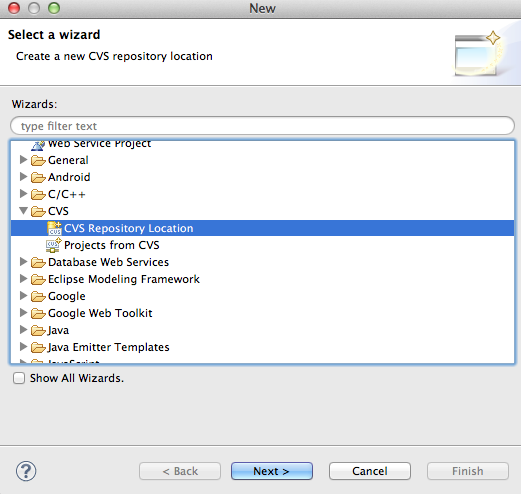
After successfully installing the Eclipse CVS Client, from the Eclipse menu, select New > CVS > CVS Repository Location…

Or you can open CVS view by menu ‘Windows>Show View>CVS Repositories’ and new a ‘Repository Location’.



For Elicpse Juno 4.2.0:

1. File -> New -> Other
2. Select CVS -> CVS Repository Location



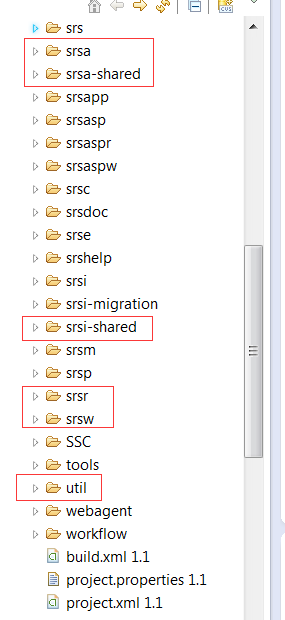
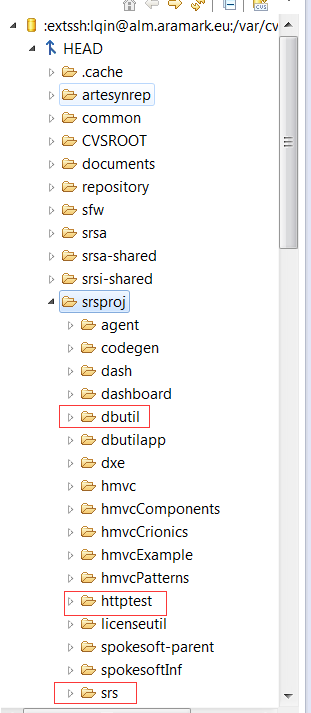
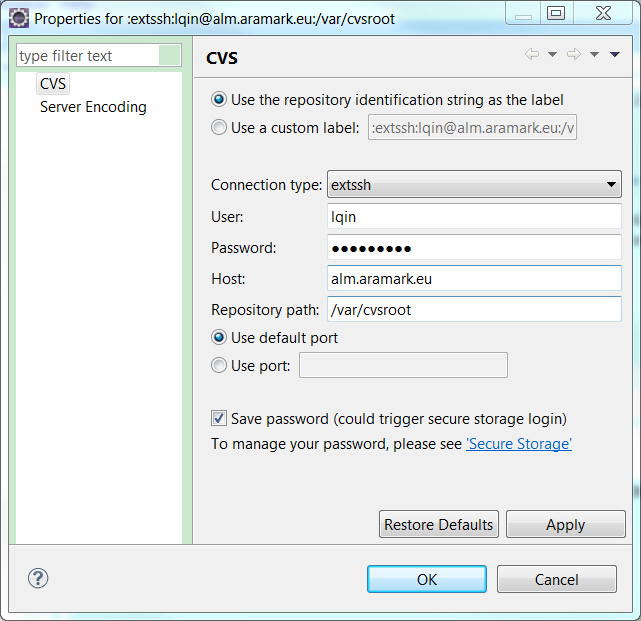
Choose 'extssh' for the Connection type

User 1:                      lqin / m2ryb6u6

User 2:               shuang / 2zuku57u

Host:                      alm.aramark.eu

Repository Path:      /var/cvsroot



RealTime FM and Related Project Check Out

Create a new Eclipse workspace and check the following projects into it.

The following is a listing of projects that should be checked out from CVS:

**RTFM**

*common*

*repository*

*srs*

*srsr*

*srsw*

*util*

**ANDROID**

1. srsa
2. srsa-shared

**INTEGRATION**

1. srsi
2. srsi-shared

**MySQL**

1. dbutil

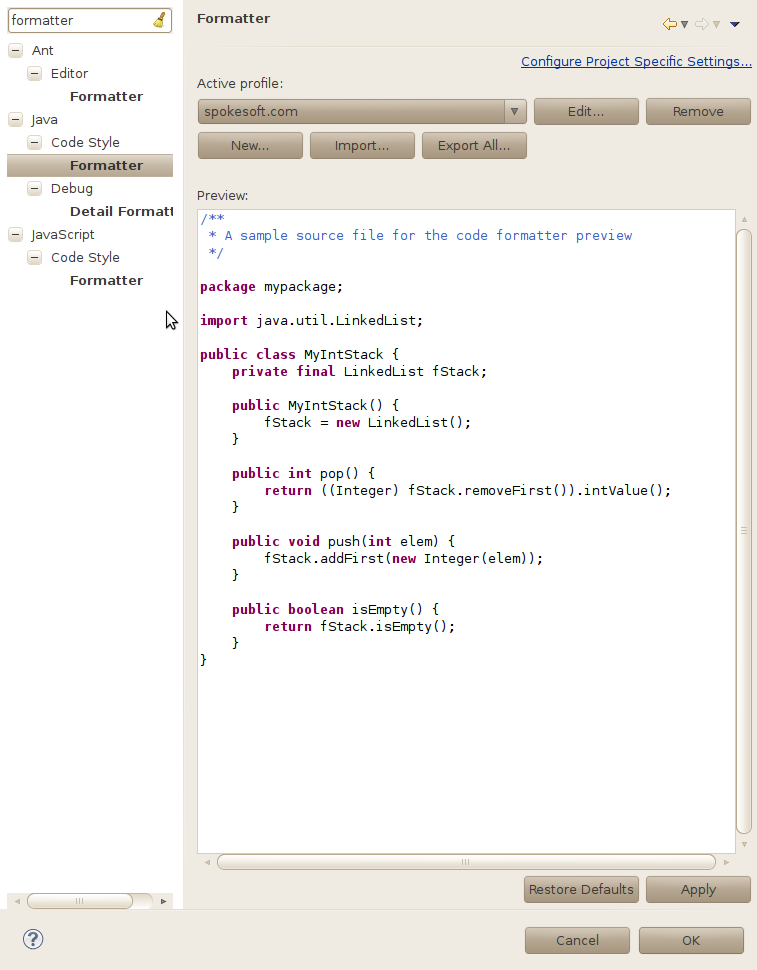
**Eclipse Coding Standard**

Developers \*\*MUST\*\* conform to the ARAMARK coding standards as defined in the configuration settings in file ‘spokesoft-coding-standard.xml’.

The following are the steps required to configure Eclipse to meet these requirements.

Code Style

1. From within the ‘common’ project, locate the following folder ‘/common/devel/eclipse.org/standards/’ in which the file ‘spokesoft-coding-standard.xml’ can be found.
2. Open Eclipse Preferences under ‘Windows -> Preferences’.
3. Using the Preferences ‘Search’ facility (*located on the top left corner of the* ***Preferences*** *window*), search for ‘formatter’.
4. Under ‘Java -> Code Style - > Formatter’, click the ‘Import’ button and select the
5. ‘spokesoft-coding-standard.xml’ file (via file system), as per screenshot below.
6. Click ‘OK’ to apply the new change.



Code Save Action

1. Open Eclipse Preferences under ‘Windows -> Preferences’.
2. Using the Preferences ‘Search’ facility, search for ‘save action’.
3. Under ‘Java -> Editor -> Save Action’, follow the settings defined in the next several screens to ensure that the setting are configured correctly.
4. Click OK, once complete

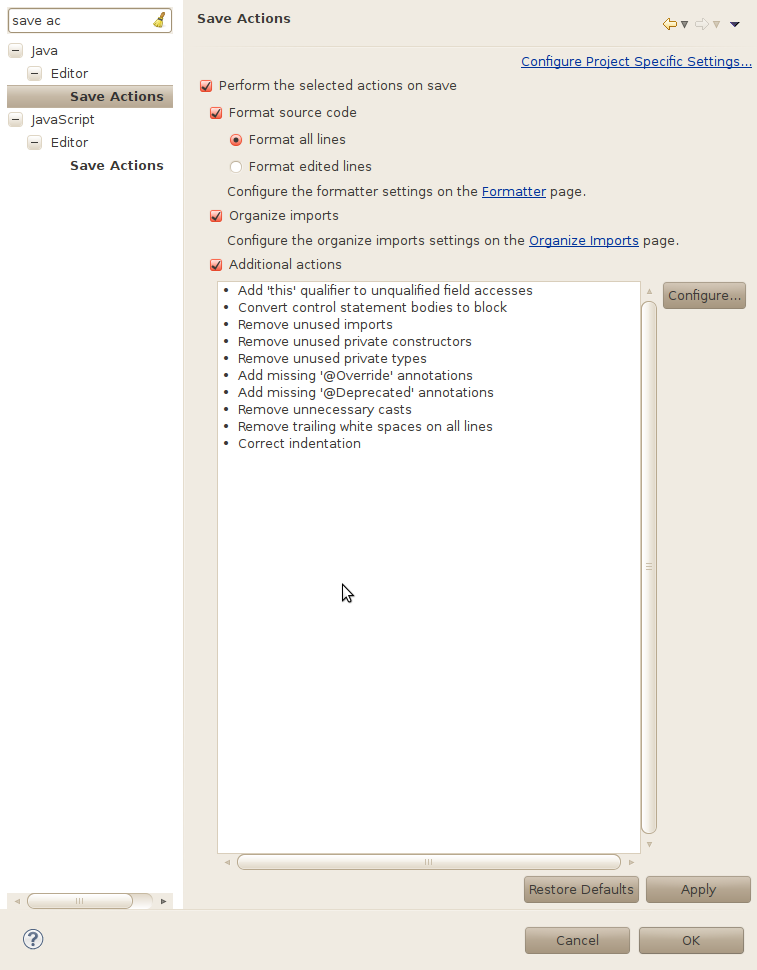
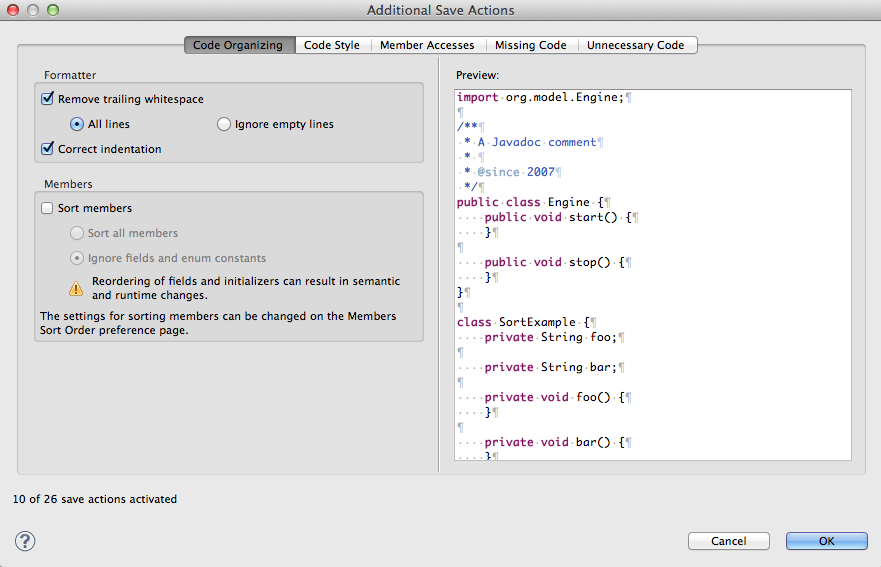
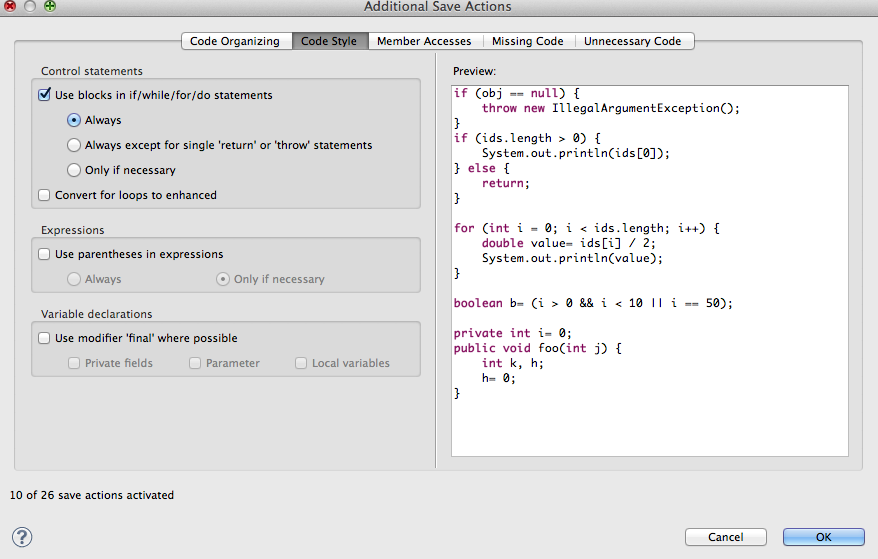
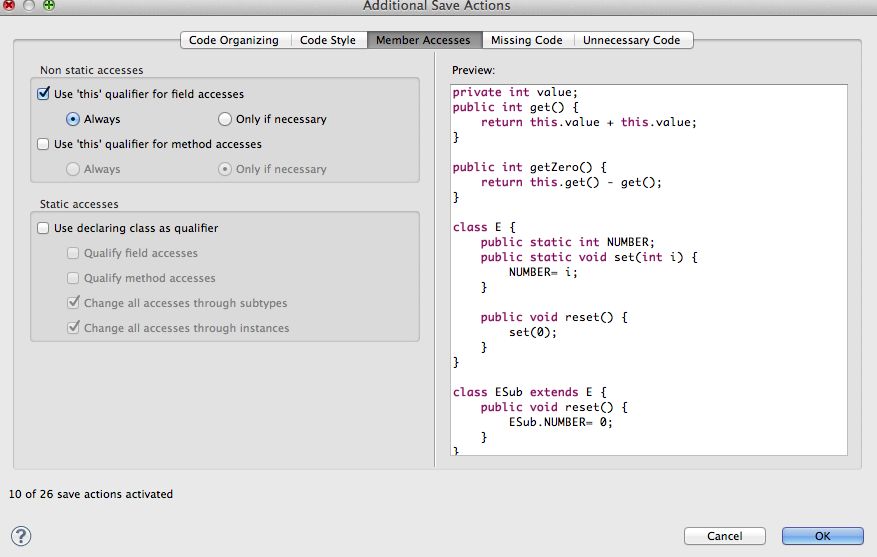
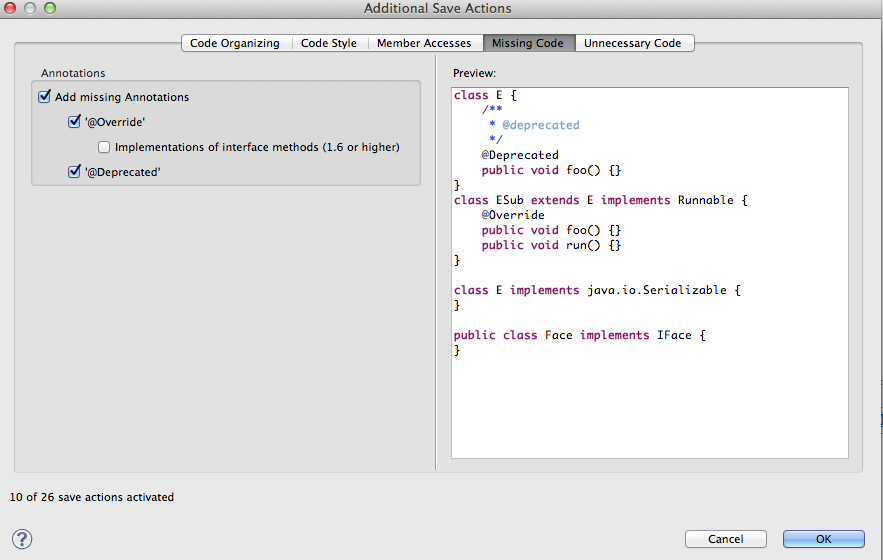
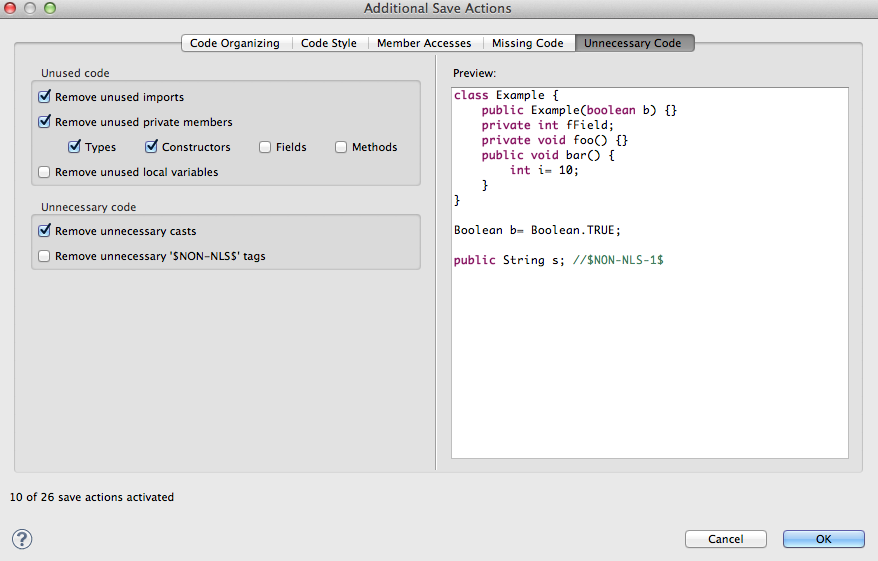


Fig: Save Action for Eclipse Java Preferences

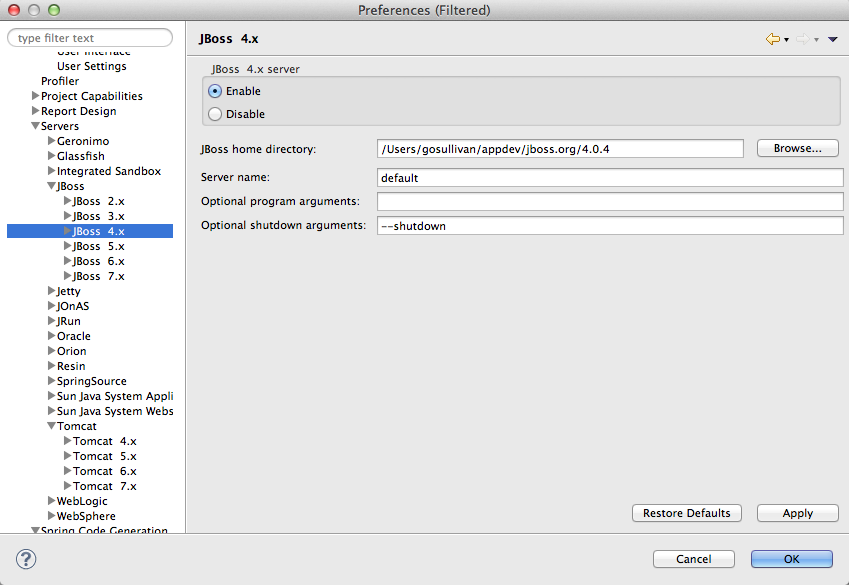
Click on the Configure button…

The following are the dialog boxes required to configure the settings for the Save Action.



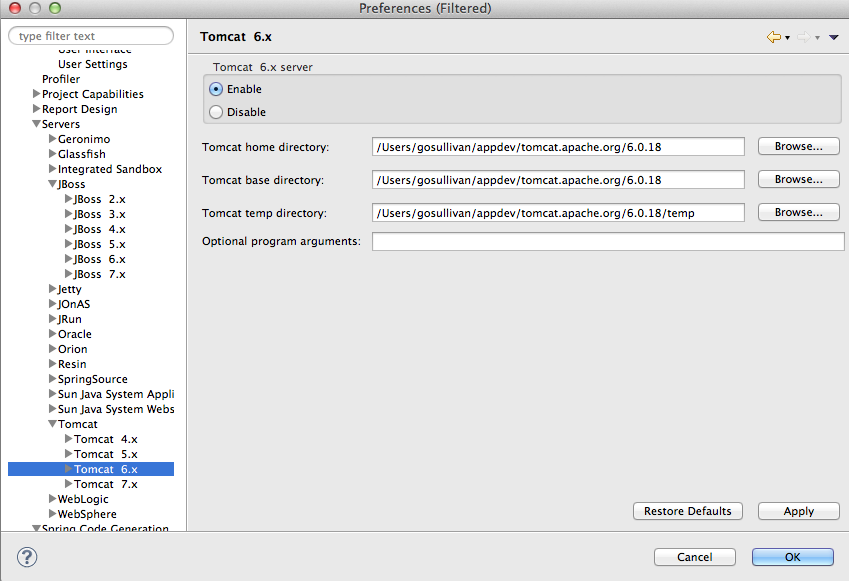
Configure JBoss Server

* From the Eclipse menu, select 'Windows > Preferences'
* On the Left Hand Side Navigation Tab under 'Servers  > JBoss 4.x', configure the server as follows:
  + JBoss Home Directory:     <jboss\_install>



Configure Apache Tomcat

* From the Eclipse menu, select 'Windows > Preferences'
* On the Left Hand Side Navigation Tab under 'Servers  > Tomcat 6.x', configure the server as follows
  + Tomcat Home Directory:     <tomcat\_install>



Configure Dependencies

* Extract savant\_repository.tar.gz content from <AWS\_CVS\_BUILD\_FILES> to the local HOME directory (i.e. a .savant\_repository folder containing the library dependencies should be placed off the user’s HOME directory).

Note: Dependencies will have to be addressed on a project-by-project basis, however, please reference each projects's ANT build file to determine the required dependencies.

Note: The project folder titled 'srsa' requires that the file 'Constants.java' in path '/srsa/src/com/spokesoft/mobile/android/srsa/' be modified according to the development environment's local configuration.

    /\*

     \* Configuration for development

     \*/

    /\*

     \* BASE URL DEVELOPMENT

     \*/

    staticfinal String \_HTTPS\_BASE\_URL\_DEV = "[https://<<ENTER\_CURRENT\_NETWORK\_IP\_ADDRESS>>:8443/srsp](https://192.168.1.103:8443/srsp)";

    /\*

     \* AWS URL

     \*/

    Static final String \_HTTPS\_BASE\_URL\_AWSIE = "<https://ws.aramark.eu:8443/srsp>";

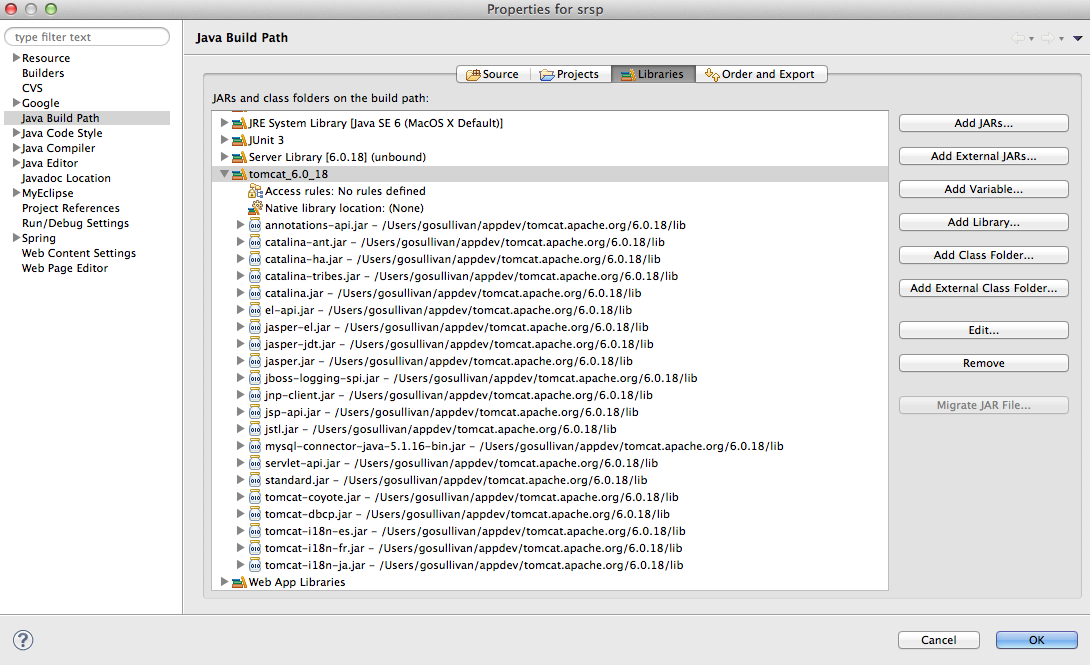
Note: The project folder titled 'srsp' requires the additional user/server library that points to the <tomcat\_install>/lib folder. Add this reference dependency manually.

Right click on the srsp project

Select **Properties**

Click on **Add Library**

Select **Server Runtime**



Configuring ANT

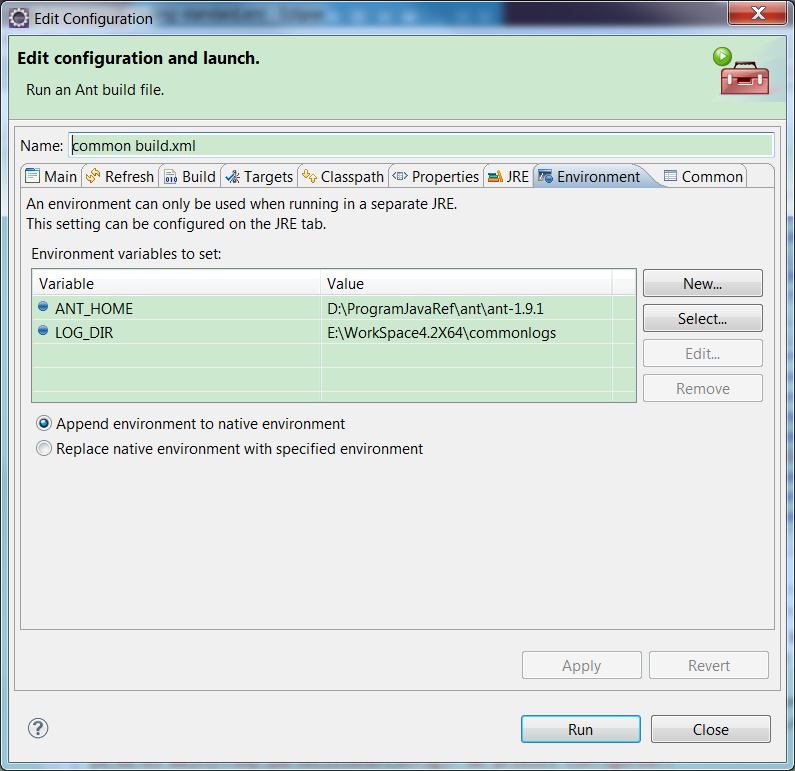
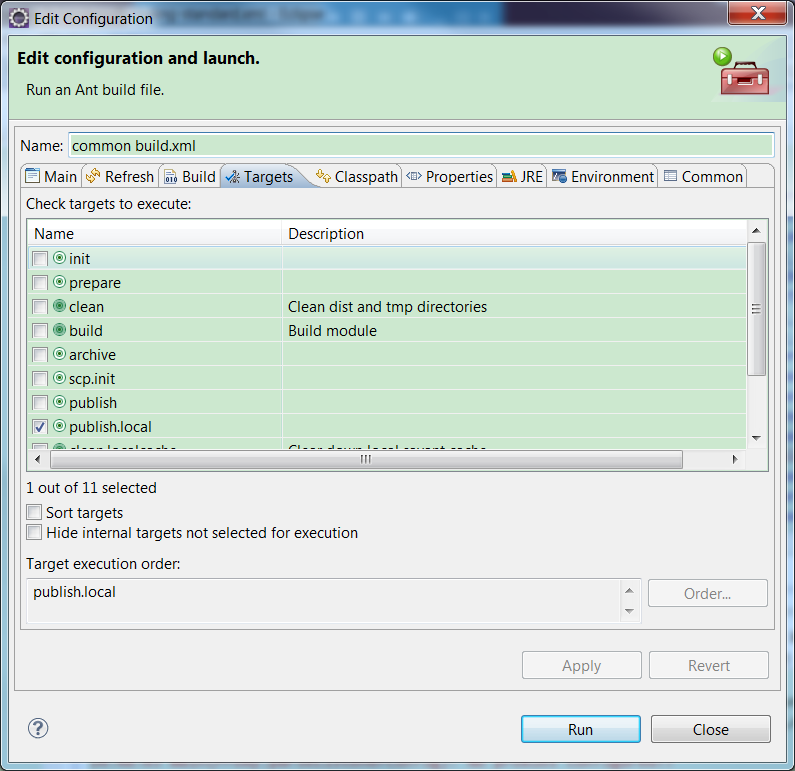
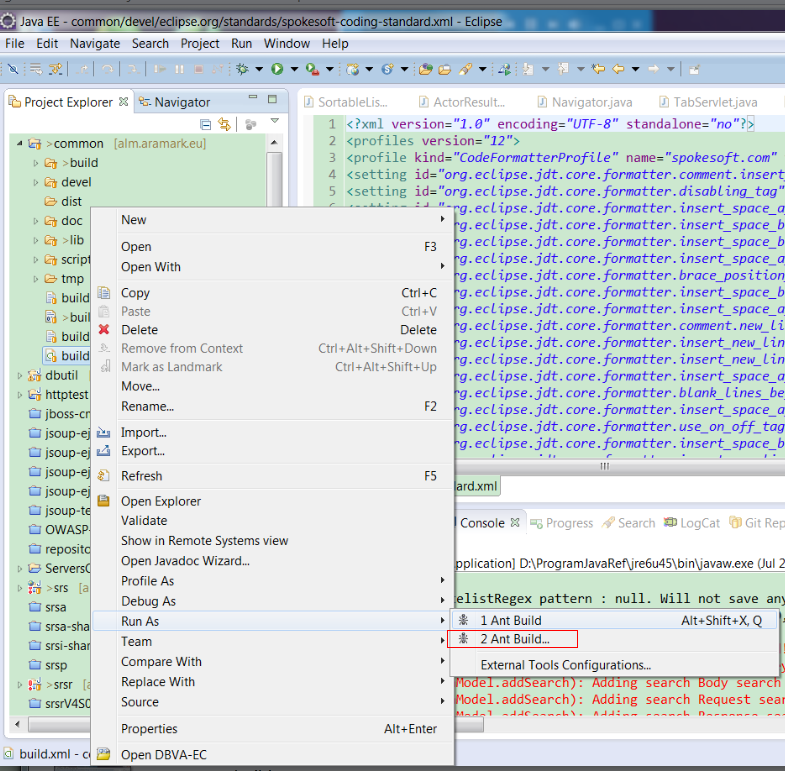
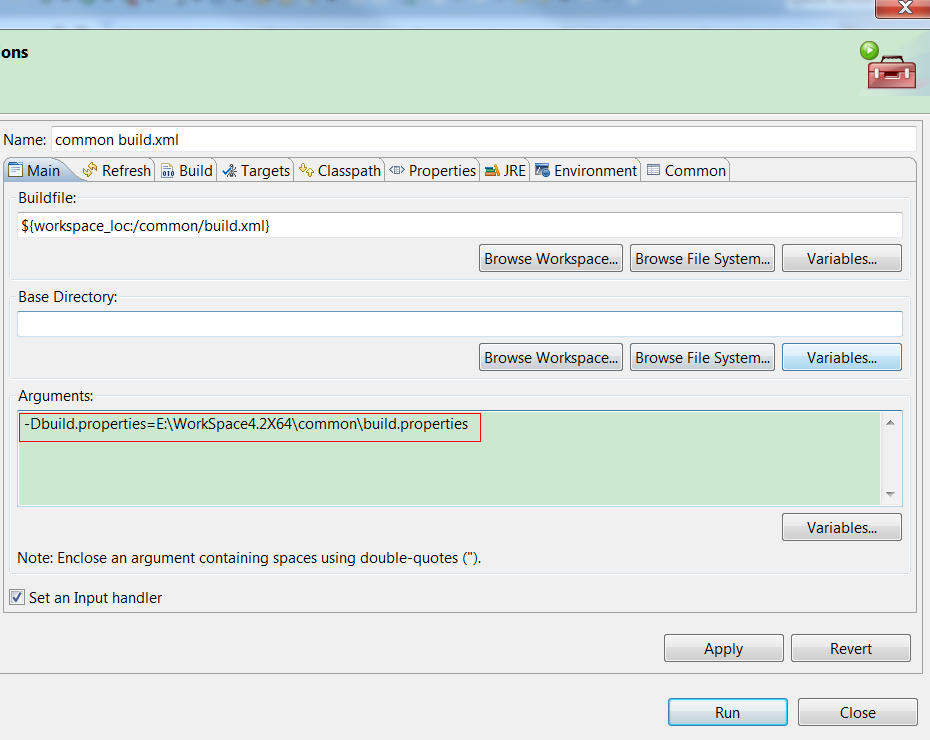
* For the following project folders, configure Ant as per screenshots provided below:
  + Common
  + Srs
  + Util
  + dbutil

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* common \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Right click on “common/build.xml”, run as “Ant Build...” to config ant.

Select tab “Targets”, ”publish.local” is selected.

Select tab “Environment”, set variable “ANT\_HOME” and “LOG\_DIR”.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* srs /projectBuild.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Click on **Run** on the menu bar

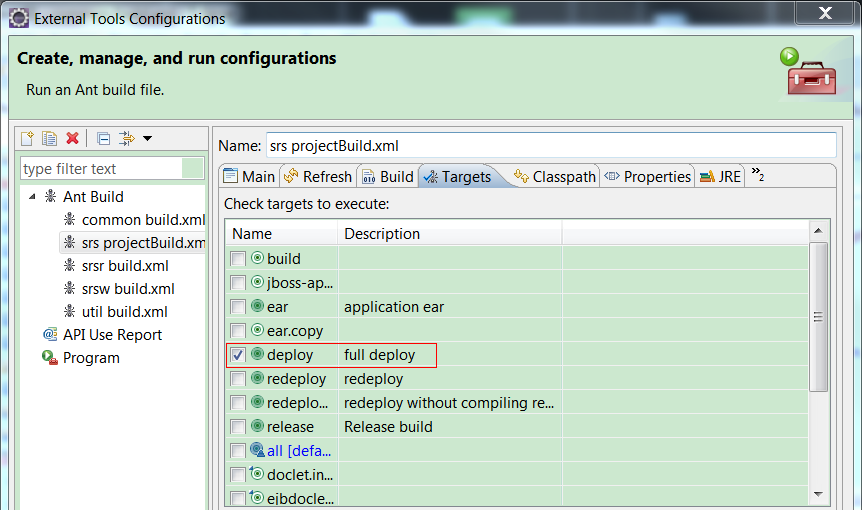
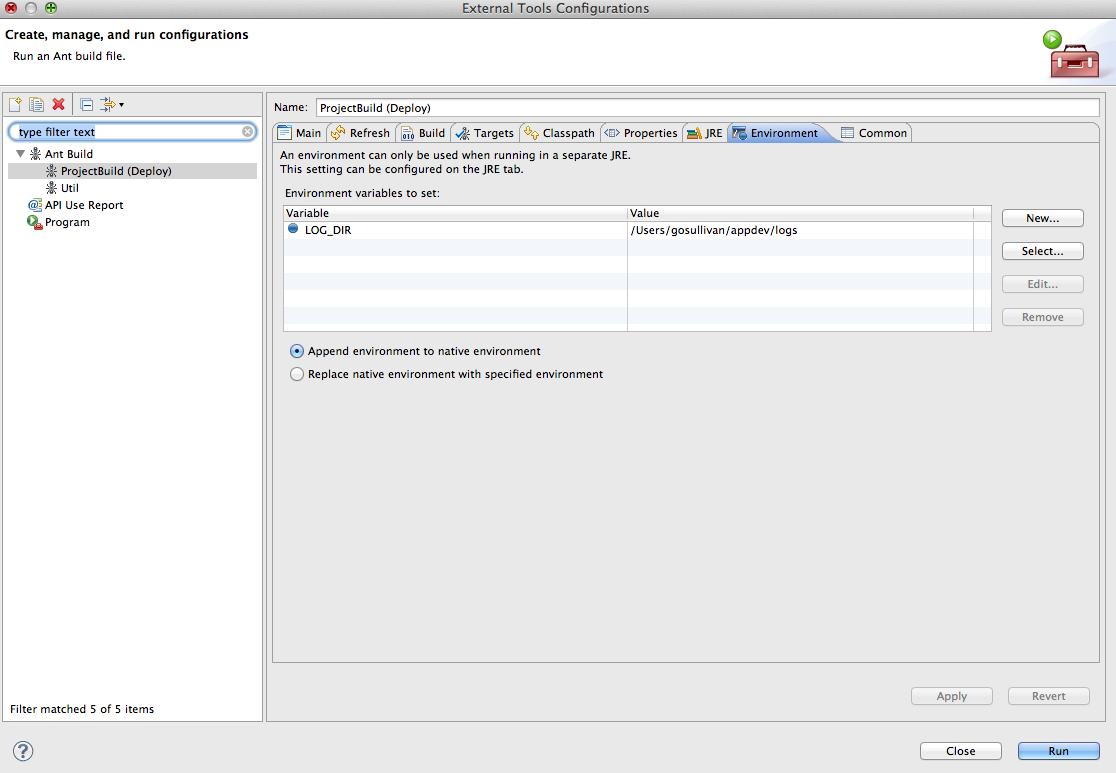
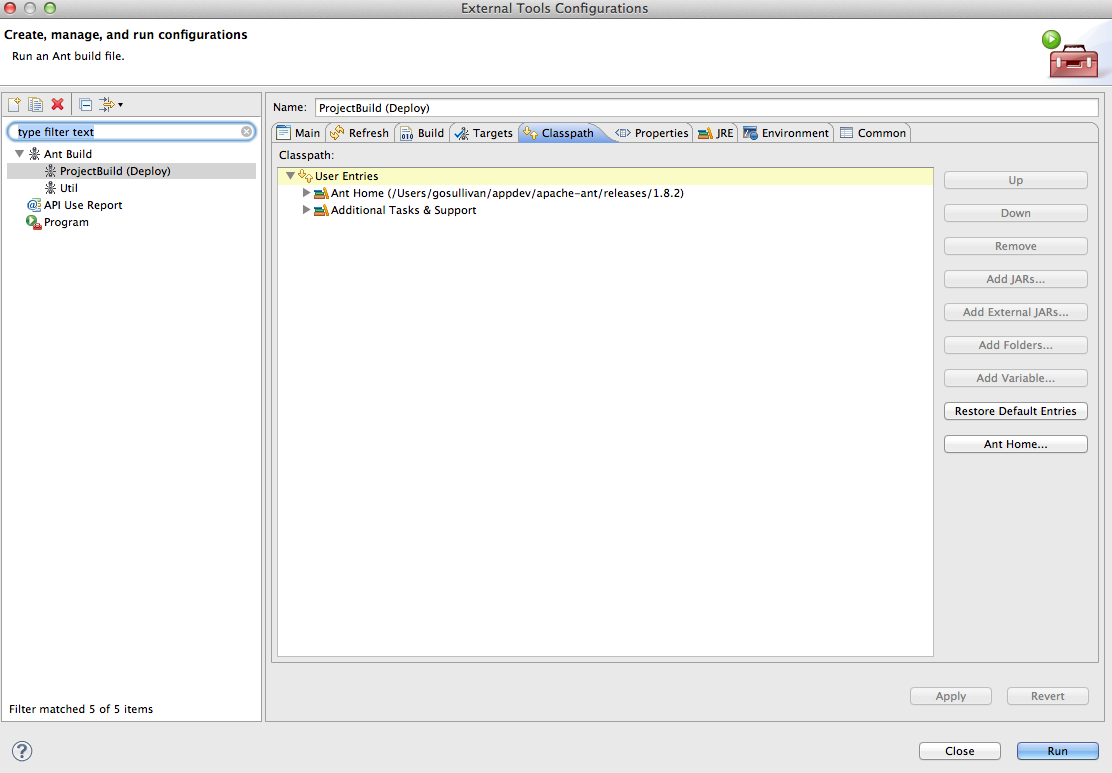
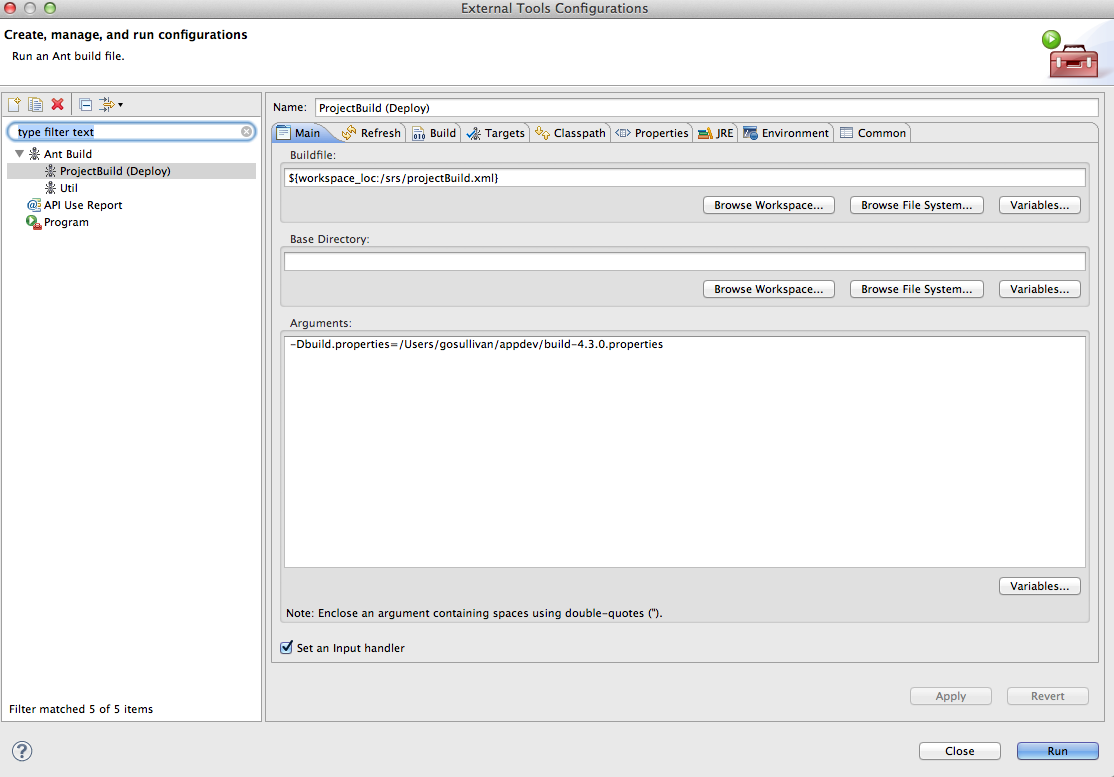
Select **External Tools**

Select **External Tools Configurations**

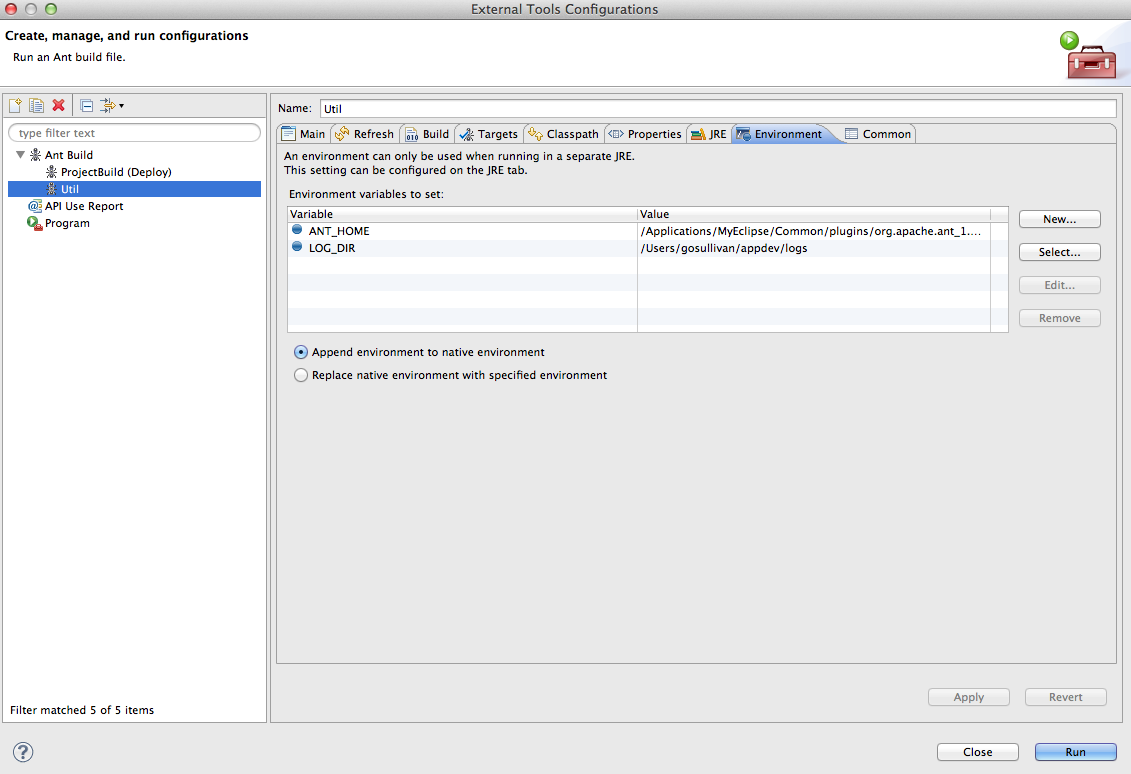
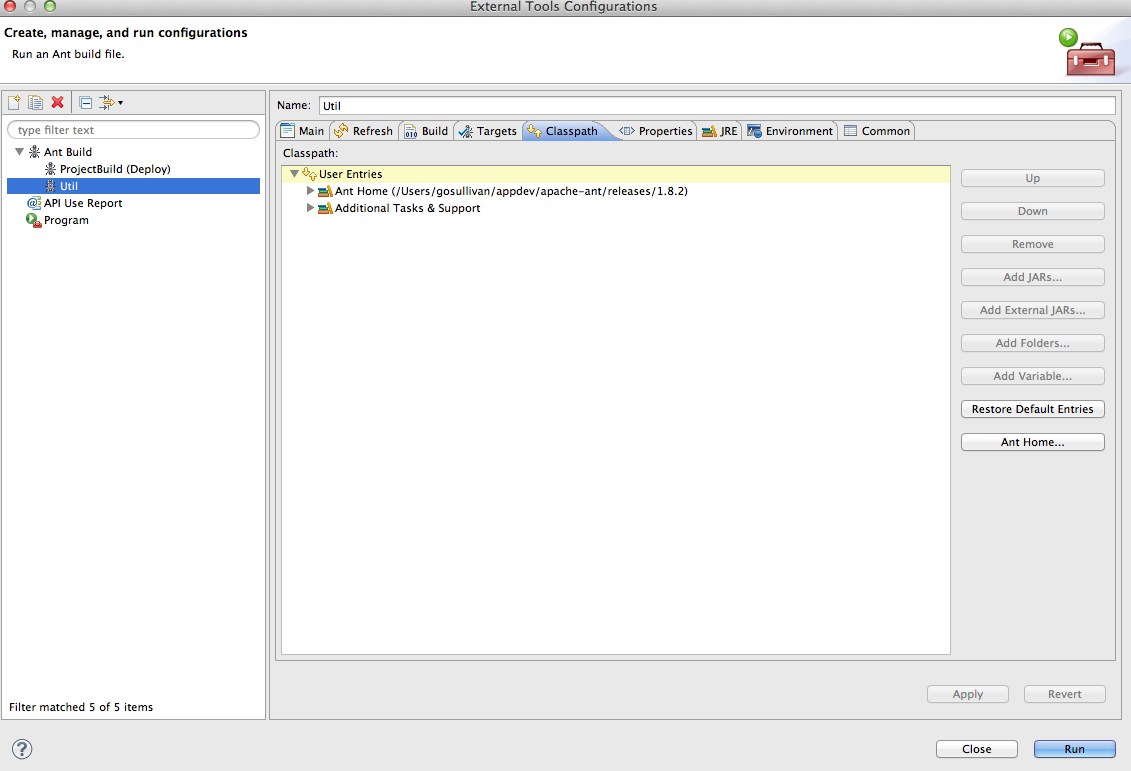
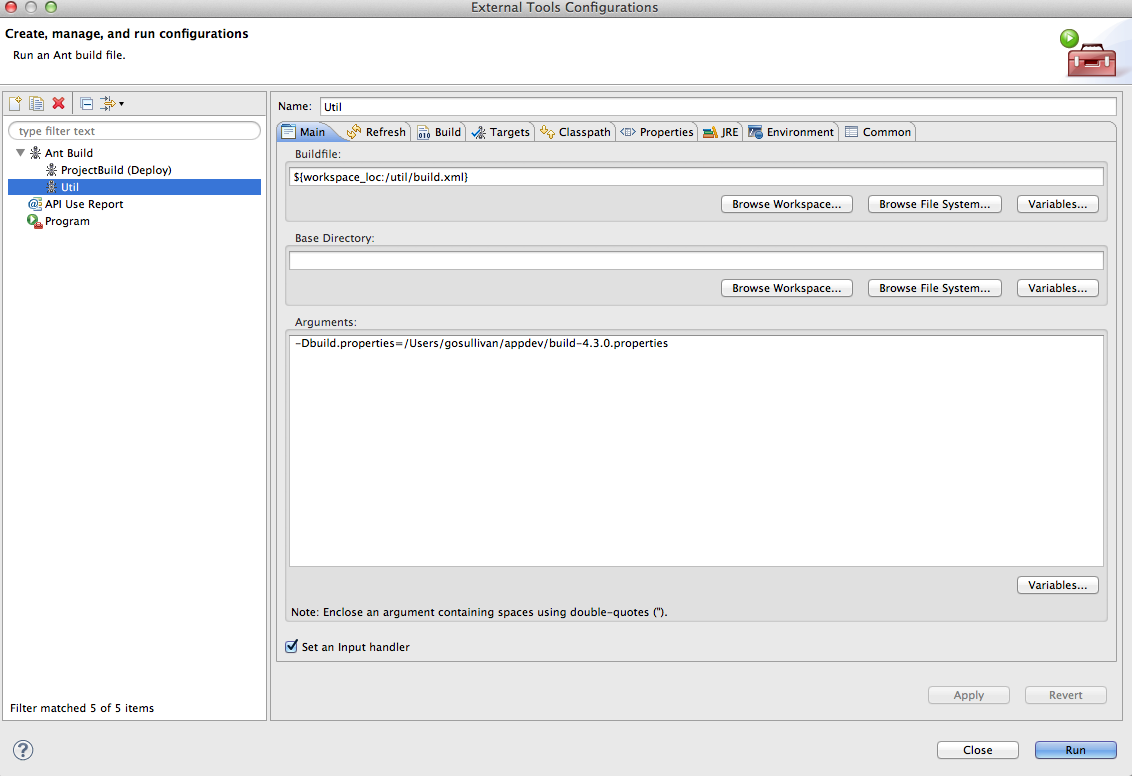
Click **Ant Build**

Click the **New** button (white document w/ a yellow + sign)

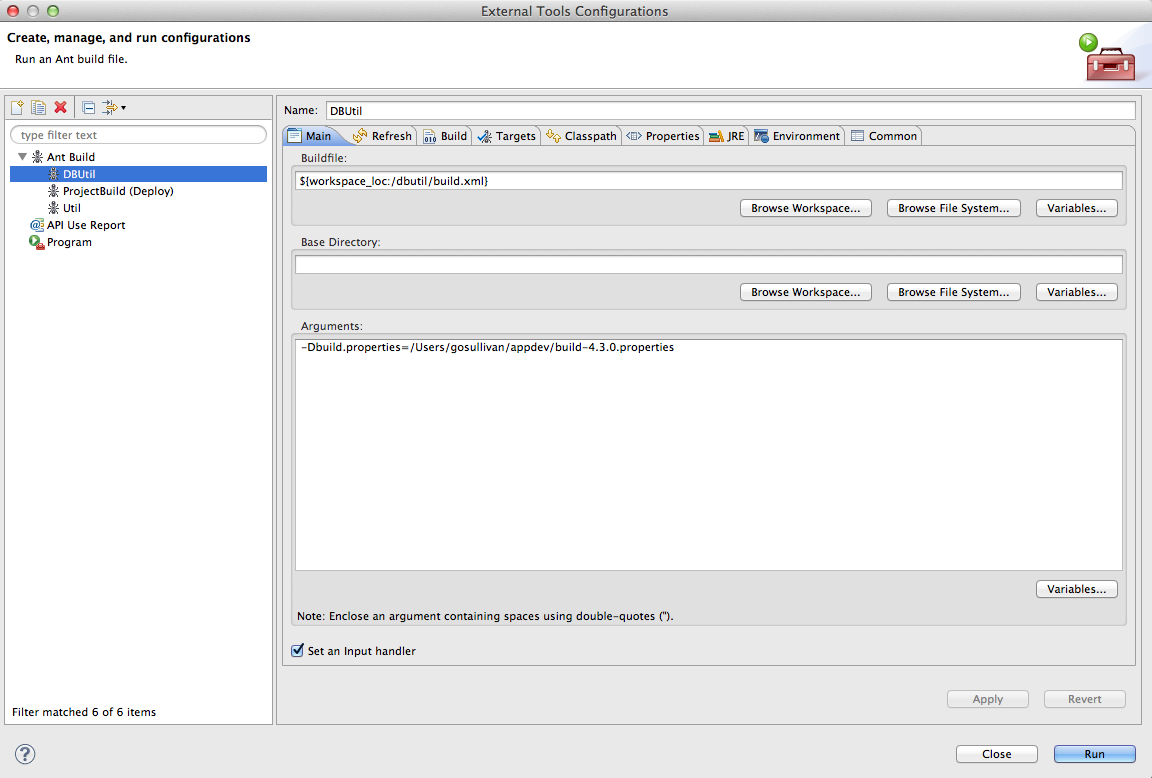
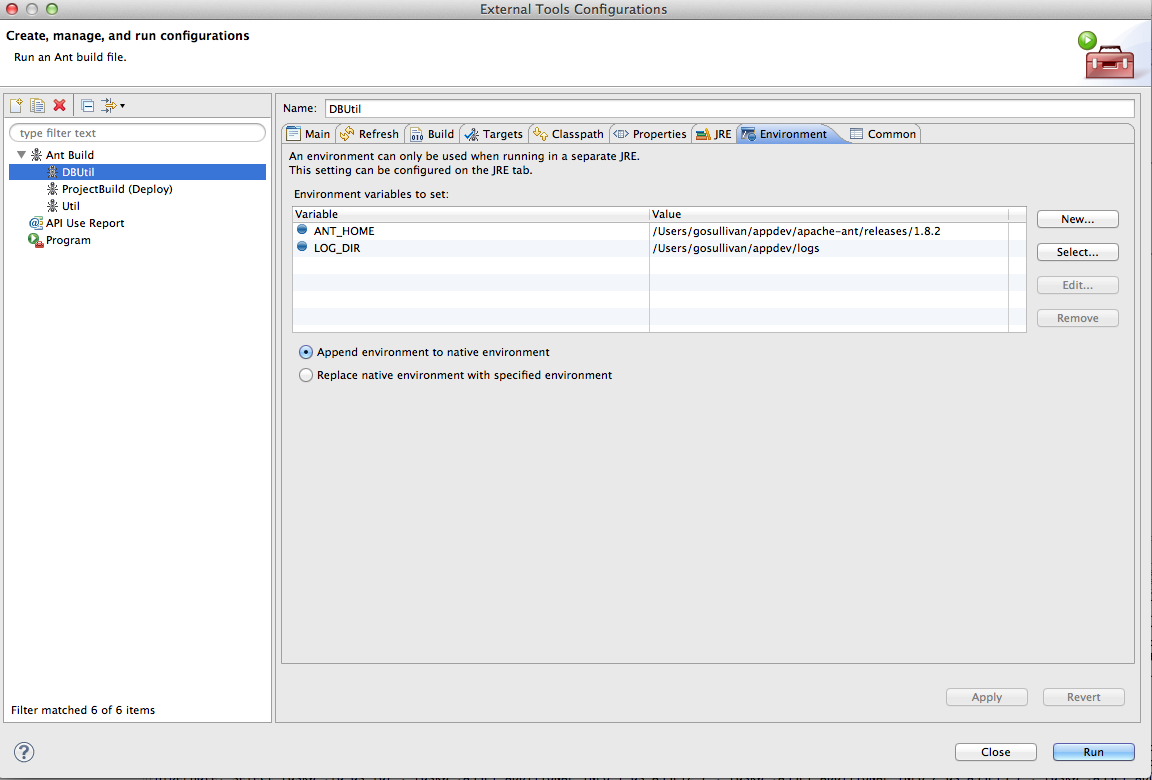
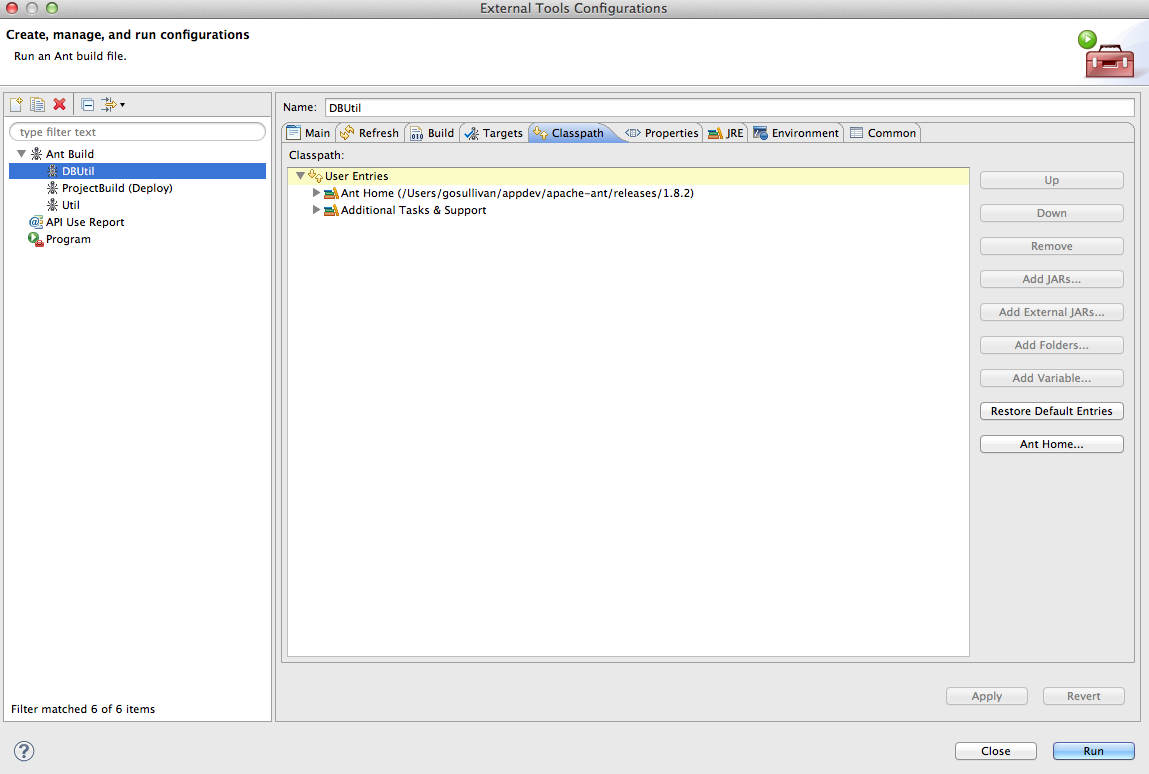
Type in info as shown in the screenshot



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Util \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DBUtil \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Configure Android Development Tools Plugin

To add the Android Development Tools (ADT) plugin to Eclipse, follow the instructions provided in the following link:

<http://developer.android.com/sdk/installing/installing-adt.html>

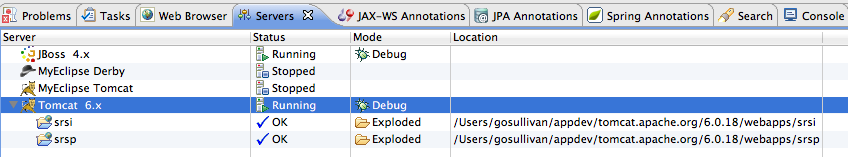
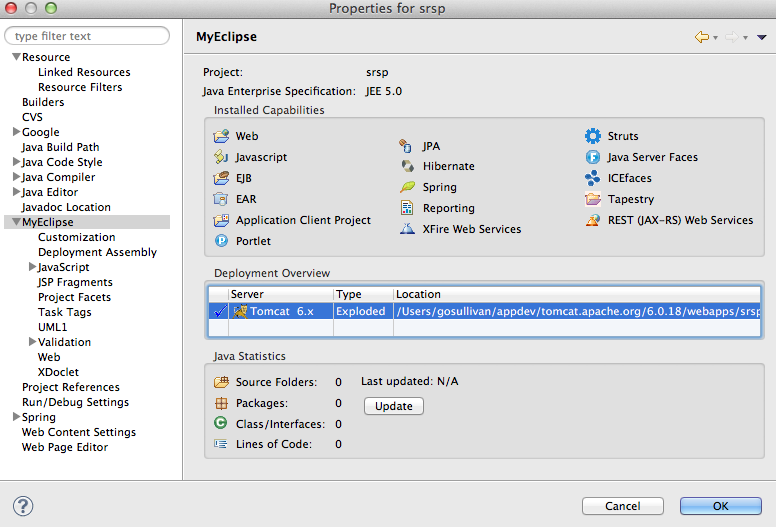
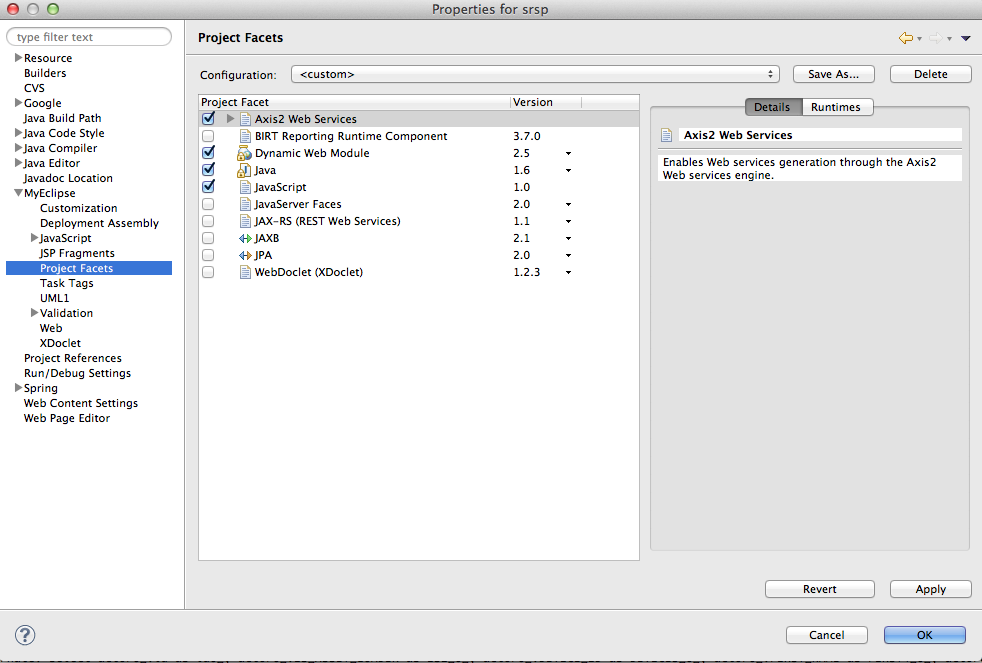
Note: The plugin requires that the Android SDK be installed on local machine.

Configure Project from Eclipse with ADT

The ‘srsp’ (Taskman) and ‘srsi’ (Enterprise) projects must be deployed to run under Tomcat as per screenshots below.

For each of the projects listed above (i.e. srsi and srsp), complete the following steps:

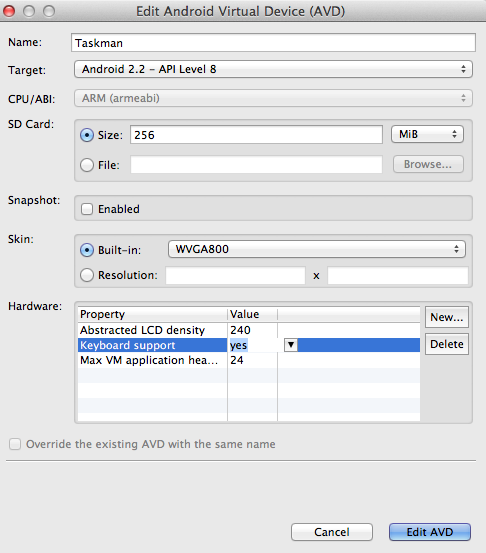
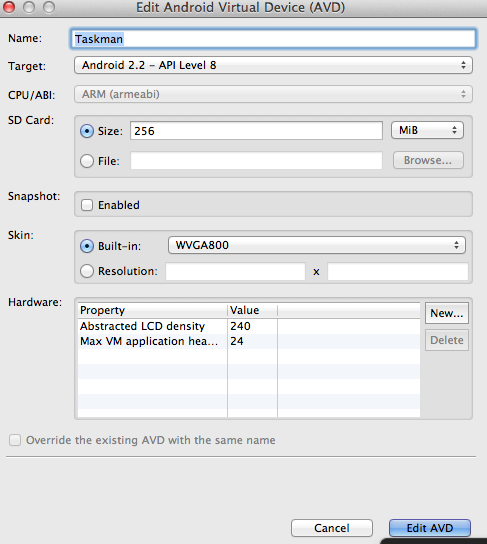
1. Right-click the project. In the context menu, select ‘Add web project capabilities’ (or similar verbiage) to allow the project source folder to find the Web Content sources.
2. Viewing the Properties menu option for the project, the following screens will display (Note: screenshots specific to MyEclipse install).
3. Dynamic Web Module should be set to version 2.5. This can be set via the dropdown context.



Configure Android Virtual Device (AVD)

To correct an issue whereby the emulator does not receive input via the keyboard, the following steps are shown to rectify the issue.

* In Eclipse, select WIndow > AVD Manager
* In the dialog box that appears, highlight 'Taskman' virtual device and select 'Edit'.
* Under 'Hardware', select 'New'.
* Under 'Property'. select 'Keyboard Support', click 'OK'.
* Under Hardware, with new property added, change the 'Value' column from 'no' to 'yes'.
* Highlight to another option under 'Hardware' to allow new value of 'Keyboard Support' = 'yes' to take effect.
* Click 'Edit AVD' to return to previous screen.
* Restart emulator for new settings to take effect.



RealTime FM Configuration

Log into the local copy of RealTime FM via

UserName: admin

Password: v3ct0r

Server: [http://localhost:<port\_number>](http://localhost:8080/)

(for example, [http://localhost:8080](http://localhost:8080/))

Add New User to RealTime FM

1. Navigate to System Admin > General Maintenance > User Management
2. From the Tab Menu, select ‘Add New User’
3. Add a new test user as per the available fields.
4. Click ‘User Roles’
5. Select All Roles
6. Select ‘Staff’ and make sure ‘All Accounts’ is selected. Click ‘Update and Return to Details’
7. Click the ‘Customer’ button. Select ‘Aramark/Spokesoft’ as the customer account.
8. Click ‘Save’.

Add New Ticket to RealTime FM

1. From the Home Page. select ‘Tickets and Tasks’.
2. Select ‘Raise Ticket For Customer’.
3. In the field labelled ‘Login Name’, enter ‘aflanagan’ and then click the ‘Search Users’ button at the foot of the form.
4. A screen displaying the ‘Name’ of ‘Aileen Flanagan’ should display with a ‘Select’ button over to the right of that row. Click the ‘Select’ button for ‘Aileen Flanagan’.
5. A new ticket for customer ‘Aileen Flanagan’ shall display. Add a ‘Description’ and click the ‘Raise New Ticket’ button.
6. The new ticket will be generated, displaying a list of Tabs for
   1. Next Action
   2. Conversation
   3. Ticket Details
   4. Task
   5. New Task
   6. Existing Task
7. Under ‘New Task’ Tab, you should now have the ability to assign the ‘Resolver’ field to the new user assigned in the previous section.
8. In order to assign a resolver, select the ‘Assign Resolver’ button.
9. In the ‘Name’ field, enter new user name and click ‘Search’.
10. When list of available resolvers display, click the new user added in previous steps.
11. Finally, click ‘Generate New Task’ to create task and send it to Taskman.

Appendices

Troubleshooting

New Tasks generated in RealTime are failing to display on Taskman application.

Ensure that the IP address configured in Constants.java is set correctly. See **Configure Dependencies.**

Clearing/Resetting Taskman Android Application

Within the simulator, select Home -> Settings -> Applications.

Locate and highlight the Taskman application

Within the Taskman application settings, select ‘Uninstall’ to remove Taskman from the emulator.

Restart the emulator.

You will be prompted to log in to the Taskman application. Since these credentials have been removed, they need to be re-added through RealTime FM. The following steps indicate how to ‘resend’ those credentials to Taskman

Resend Taskman user credentials

Open RealTime FM and navigate to Home Page -> System Admin -> General Maintenance -> User Management.

Locate your current user’s name via a Search.

Within the User Profile, deactivate the user by deselecting the ‘Active?’ checkbox. Click ‘Save’ to save the changes.

To reactivate the user, search for the user name again (this time with the ‘Active’ field dropdown set to ‘No’).

Reactivate the user by selecting the ‘Active’ and click ‘Save’.

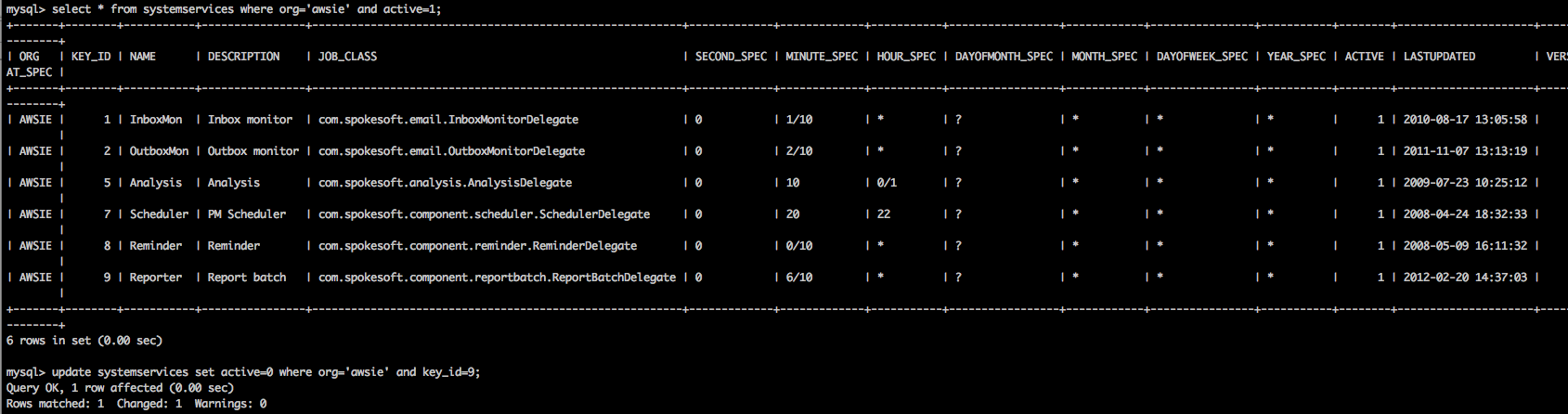
\*\*\*\*Important: Ensure that the ‘Device ID’ field is populated.\*\*\*

This will have the effect of sending the user information to the Android emulator.

RealTime FM Reporter Service

View existing service status by opening the following URL.

<http://localhost:8080/srs/ServiceManager>



**Build version number**

**Overview**

Each release of RealTime /FM is given a specific version number. This is an essential element of the development process which ties the product roadmap, QA, UAT, production, issue management, test planning and execution together.

At time of writing, the following RealTime /FM versions are in scope

* 5.1.3 Current version in production
* 5.2.0 Version undergoing QA and UAT
* 5.3.0 Next target version under development

**Versioning**

The following are the version convensions used in RealTime /FM:

* **Release Branch**: major\_minor (rb\_5\_2)
* **Release**: major*minor*build (rel\_5\_2\_0)
* **RealTime /FM**: major.minor.build (5.2.0)

**Development environment configuration**

The following process should be followed at the beginning of each release cycle to prepare the developers environment for the next target version of RealTime /FM.

**1. build.properties**

The build.properties file is used to configure the RealTime /FM build system. The file should **not** be checked into CVS as it's contents are developer, environment and release specific. The build.properties file should be located **outside** of the eclipse RealTime /FM project workspace. The build.properties should include the major.minor.build version in it's name for clarity.

Please note that currently there is no requirement to increment the build number of major.minor.build for each build.

The following shows build.properties files over time:

The developer can choose the location of the directory where the build.properties file(s) are located. The only criteria is this location should be outside of the eclipse RealTime /FM project workspace.   
For the purposes of illustration, directory $HOME/devel/build is used.

* Copy previous build.properties to new file build-major.minor.0.properties. For the next RealTime /FM target version under development, this will be build-5.3.0.properties
* Edit build-major.minor.0.properties for the next target version under development
  1. Update product.version property to next target version major.minor under development (This is used in jar, war and build task versioning)
  2. Leave product.build property at version 0 (This is used in jar, war and build task versioning)
  3. Update product.release to the next target version label (This is *only* used to correctly label the classcom.spokesoft.component.license.Version.java which the RealTime /FM user can see from the RealTime /FM home page via the 'About RTFM' menu option)
  4. Update dir.srsroot to the location of the eclipse project workspace location
  5. Update dir.utilroot to the location of the eclipse project workspace location

The following segment shows the relevant properties for build-5.3.0.properties file. The complete file is attached.

###############################################################################

# SRS master build properties. #

#----------------------------- #

# Modify for local development environment. #

###############################################################################

# product major and minor version number

» product.version=5.3

# current build number

» product.build=0

# release label

» product.release=5.3.0

# Development root directories

» dir.srsroot=/home/mcrowley/devel/workspace/srsproj-mainline

» dir.utilroot=/home/mcrowley/devel/workspace/srsproj-mainline

…

**2. Eclipse ant build.properties configuration**

* Configure Eclipse to use the build-major.minor.0.properties file for the next target version under development. This configures environment variable build.properties for the eclipse workspace
  1. The ant, global property to update is build.properties

The following shows the relevant eclipse preference configuration

**3. Savant and the common project**

Savant is used within the RealTime /FM, Ant build system to manage build dependencies in a manner similar to Maven.

Build dependencies include:

* Third party libraries
* Build task common targets (Compile, Clean, Jar, etc…)

Build task common targets are versioned using the same numbering conventions as RealTime /FM. This is subject to review.

To prepare third party libraries and build task common targets for the next target version under development, target publish.local in build file ${workspace\_loc:/common/build.xml} must be executed.   
This publish all required dependencies to the .savant\_repository directory in the users home directory.

The following shows the build file to execute:

The following shows the build file target to execute:

**4. RealTime /FM build targets**

The following srs/projectBuild.xml targets should be executed depending on requirements:

* deploy **Default** Full RealTime /FM build
  + Build copied to JBoss deploy directory as configured in build.properties : dir.deploy property
* redeploy RealTime /FM build
  + Excludes the XDoclet annotation processing (Faster! but EJB and Web XML changes not updated)
  + Build copied to JBoss deploy directory as configured in build.properties : dir.deploy property
* release Full RealTime /FM release build
  + Displays release information, compiler settings
  + Request database type input
  + Includes com.spokesoft.component.license.Version update
  + Build copied to JBoss deploy directory as configured in build.properties : dir.deploy property

**5. MobileGateway**

There is currently no Ant build.xml for the MobileGateway. The MobileGateway is built using the eclipse build functionality.

This will likely change in the coming months and this process will be updated accordingly.

Class com.spokesoft.mobile.amg.Constants should be updated for each release to increment the application version number.

For the next target version under development:

* Constant APPLICATION\_VERSION\_NUMBER constant will be 5

**package** **com.spokesoft.mobile.amg**;

/\*\*

\* <p>

\* Constants

\* </p>

\*

\* @author Brendan Lawlor

\*

\*/

**public** **class** **Constants** {

/\*\*

\* Application version number

\*/

» **public** **static** **final** **int** APPLICATION\_VERSION\_NUMBER = 5;

}

h3. 6. TaskMan

File AndroidManifest should be updated for each release to increment the versionCode and versionName attributes.

For the next target version under development:

* Attribute versionCode will be 5 (This number should equal the APPLICATION\_VERSION\_NUMBER in the MobileGateway)
* Attribute versionName will be 5.3

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.spokesoft.mobile.android.srsa"

» android:versionCode="5"

» android:versionName="5.3"

>

…

</manifest>

**7. TaskMan-shared (Shared data protocol between TaskMan and the MobileGateway)**

There is currently no Ant build.xml for the TaskMan-shared. TaskMan-shared is built using the eclipse build functionality.

This will change in the coming months and this process will be updated accordingly.

Database upgrade utility

<https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility>

* [Database upgrade utility](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility" \l "Database-upgrade-utility)
  + [Overview](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility" \l "Overview)
  + [Schema management](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility" \l "Schema-management)
  + [Migration script](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility" \l "Migration-script)
  + [Database upgrade utility](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility" \l "Database-upgrade-utility-2)

# Database upgrade utility

## Overview

DBUtil is a RealTime /FM support application for managing database upgrades from one version to the next. There are 3 elements to the application. These are:

1. Schema management
2. Migration scripts
3. Database upgrade utility

All elements are contained in the eclipse dbutil project which is located in cvs at location CVSROOT/srsproj/dbutil

## Schema management

All database changes are listed in /dbutil/webfiles/resources/schema.csv

Each row represents a single database schema operation. These are:

* CREATETABLE
* DEFINEKEY
* DROPTABLE
* ADDFIELD
* CHANGEFIELD
* DROPFIELD

The first field is the database version that the row refers to. For simplicity, the database version is now linked to the RealTime /FM release number. At time of writing the RealTime /FM release is 5.4.0 which translates to database version 540

The RealTime /FM database stores the current database version in installation.DB\_BUILD

The second field is the schema operation. The remaining columns provide arguments for the database operation.

Example row:  
  
540,ADDFIELD,actors,VARCHAR,50,,,,,LOGIN\_TOKEN

It is normal practice to group all rows relating to a table together.

**NOTE** Only open the schema.csv file in a text editor. Opening the file in Excel can render the file unusable for the dbutil application.

## Migration script

Each database version has an associated upgrade class. Upgrade classes are located in /dbutil/src/com/spokesoft/dbupgrade/

At time of writing the current upgrade class is Upgrade540.java

The upgrade class is used to perform data initialisation, migrations, adding new reports etc…

## Database upgrade utility

DBUtil is built as a enterprise archive (ear) and deployed to the JBoss deploy directory.

Once deployed, navigate to http://{jbosshost}:{jbossport}/dbutil/Dispatcher

The login details are:  
  
user: dba  
password: spokey

* Adjust the Database URL, Database User and Password to match your environment
* Enter the required database Build Number. Currently this is 540
* The default value for From Build Number is -1. The default behavior is to apply schema changes from the value ininstallation.DB\_BUILD to the value specified in Build Number. During the development cycle there may be situations where the developer needs to repeat a database upgrade. In this case installation.DB\_BUILD and Build Number will match and no upgrade will occur. Setting From Build Number to the (target build number -1) which in this instance is 539 will force the database upgrade utility to re-run all steps associated with the version number.

### Project build path

Import eclipse user libraries “RTFM\_20150505.userlibraries”

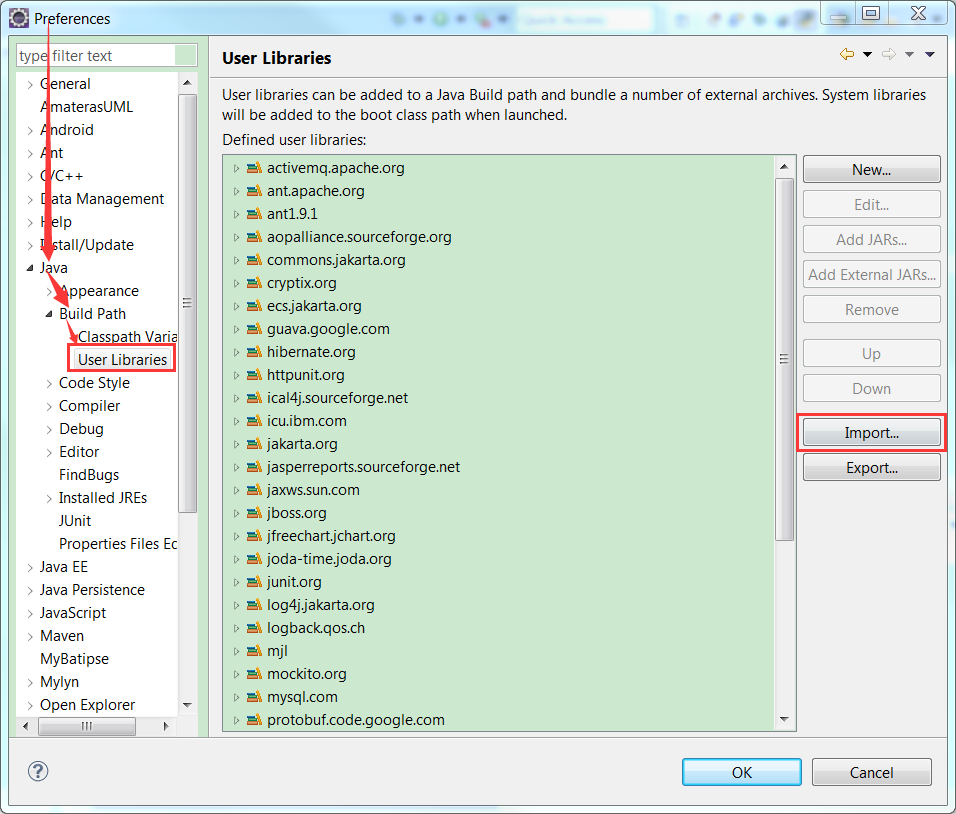
1. Replace the path of userlibraries.

Users/vince.feng -> Users/yourFirstName.lastName

D:/ProgramJavaRef/ant/ant-1.9.1 -> yourAntPath

source="mySourcePath" -> yourSourcePath

Eclipse->Window->Preferences->Java->Build Path->User Libraries->Import



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*util\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### 1 Create or replace util/.project

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>util</name>

<comment></comment>

<projects>

</projects>

<buildSpec>

<buildCommand>

<name>org.eclipse.jdt.core.javabuilder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.wst.common.project.facet.core.builder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>org.eclipse.wst.common.project.facet.core.nature</nature>

<nature>org.eclipse.jdt.core.javanature</nature>

</natures>

</projectDescription>

### 1.2 Create or replace util/.classpath

<?xml version="1.0" encoding="UTF-8"?>

<classpath>

<classpathentry kind="src" path="src/java"/>

<classpathentry kind="src" path="src/test"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/guava.google.com"/>

<classpathentry combineaccessrules="false" kind="src" path="/srs"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jasperreports.sourceforge.net"/>

<classpathentry kind="con" path="org.eclipse.jdt.junit.JUNIT\_CONTAINER/4"/>

<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.7">

<attributes>

<attribute name="owner.project.facets" value="java"/>

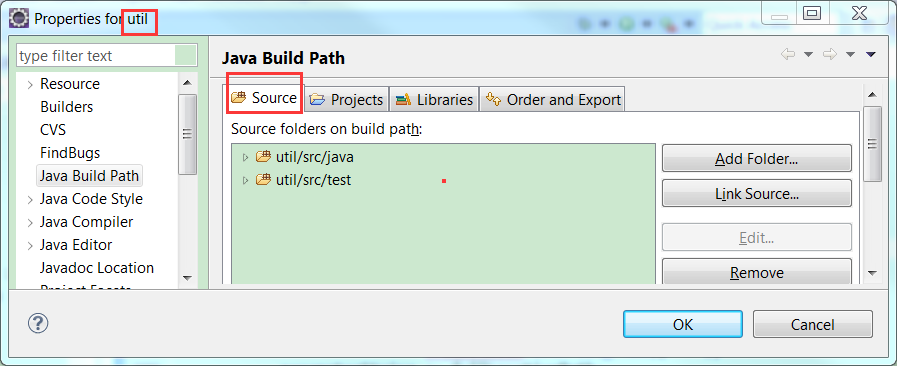
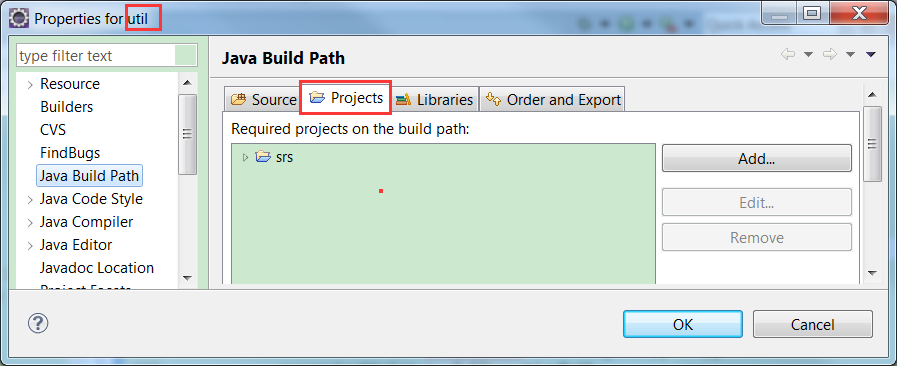
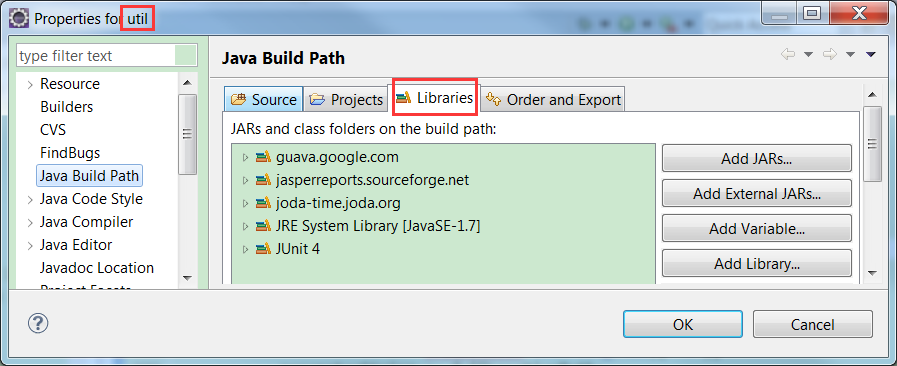
</attributes>

</classpathentry>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/joda-time.joda.org"/>

<classpathentry kind="output" path="bin"/>

</classpath>



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*srs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### Create or replace srs/.project

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>srs</name>

<comment></comment>

<projects>

</projects>

<buildSpec>

<buildCommand>

<name>org.eclipse.wst.jsdt.core.javascriptValidator</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.wst.common.project.facet.core.builder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.wst.validation.validationbuilder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.jst.j2ee.ejb.annotations.xdoclet.xdocletbuilder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>org.eclipse.jem.workbench.JavaEMFNature</nature>

<nature>org.eclipse.wst.common.modulecore.ModuleCoreNature</nature>

<nature>org.eclipse.jdt.core.javanature</nature>

<nature>org.eclipse.wst.common.project.facet.core.nature</nature>

<nature>org.eclipse.wst.jsdt.core.jsNature</nature>

</natures>

</projectDescription>

### 2.Create or replace srs/.classpath

<?xml version="1.0" encoding="UTF-8"?>

<classpath>

<classpathentry kind="src" path="src"/>

<classpathentry combineaccessrules="false" kind="src" path="/srsi-shared"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jboss.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/xstream.codehaus.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.junit.JUNIT\_CONTAINER/4"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/springframework.org"/>

<classpathentry combineaccessrules="false" kind="src" path="/util"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/commons.jakarta.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/ical4j.sourceforge.net"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/ant.apache.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/cryptix.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/icu.ibm.com"/>

<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.7">

<attributes>

<attribute name="owner.project.facets" value="java"/>

</attributes>

</classpathentry>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/guava.google.com"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/ecs.jakarta.org"/>

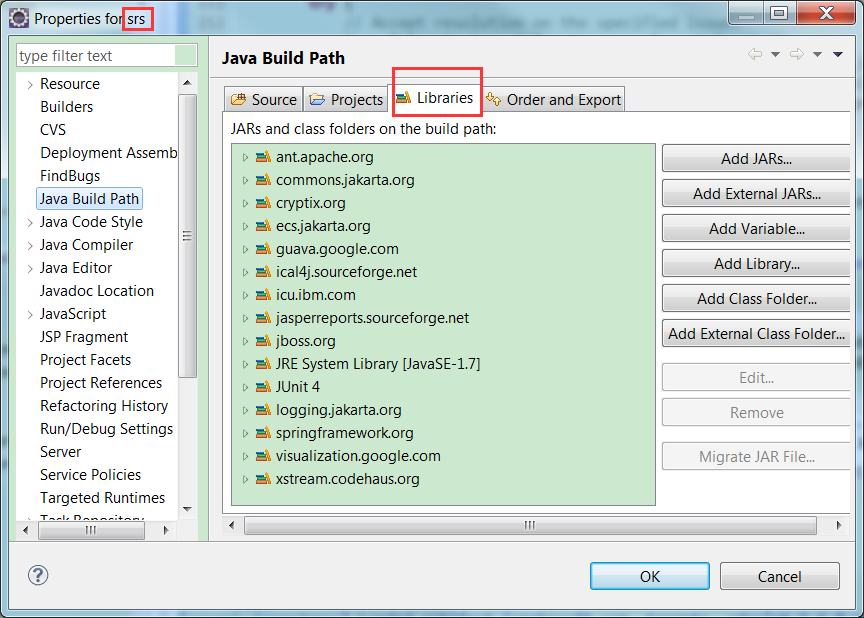
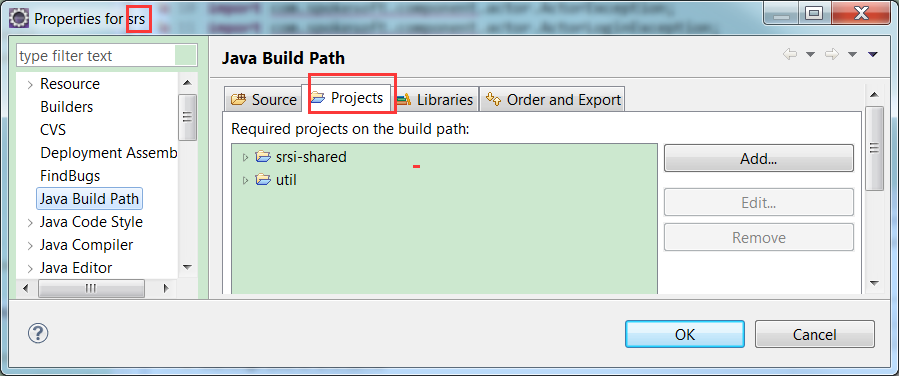
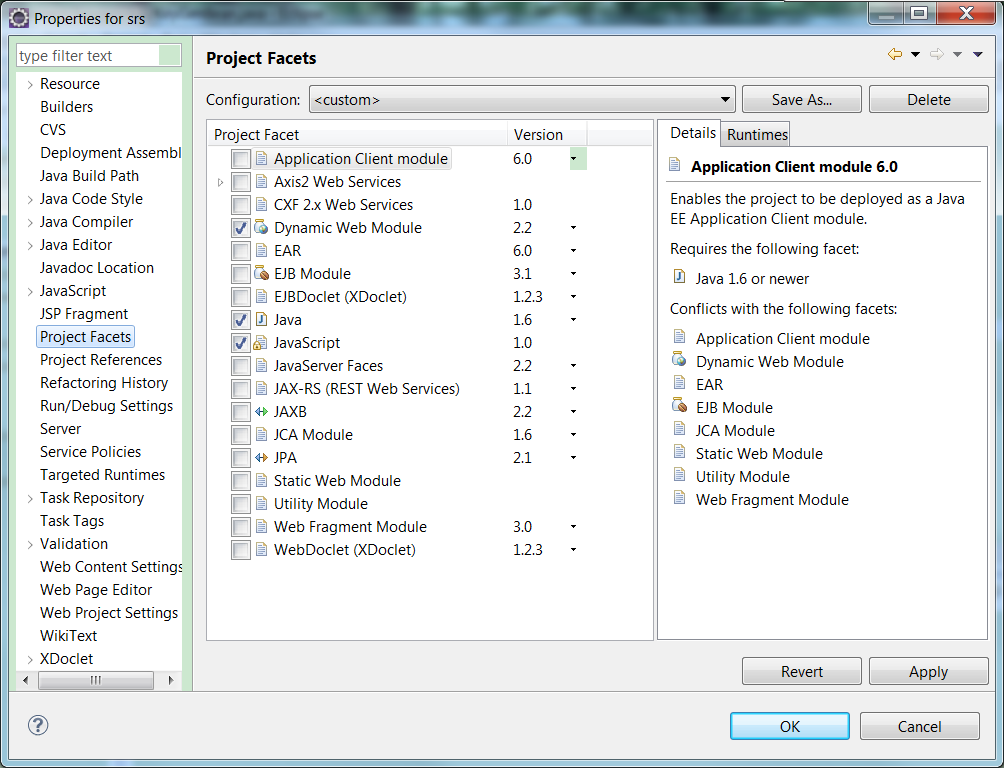
<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/logging.jakarta.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jasperreports.sourceforge.net"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/visualization.google.com"/>

<classpathentry kind="output" path="bin"/>

</classpath>



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*srsw\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### 1.Create or replace srsw/.project

<?xml version="1.0" encoding="UTF-8"?>

<projectDescription>

<name>srsw</name>

<comment></comment>

<projects>

<project>srs</project>

<project>srsr</project>

<project>util</project>

</projects>

<buildSpec>

<buildCommand>

<name>org.eclipse.wst.jsdt.core.javascriptValidator</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.wst.common.project.facet.core.builder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.jdt.core.javabuilder</name>

<arguments>

</arguments>

</buildCommand>

<buildCommand>

<name>org.eclipse.wst.validation.validationbuilder</name>

<arguments>

</arguments>

</buildCommand>

</buildSpec>

<natures>

<nature>org.eclipse.jem.workbench.JavaEMFNature</nature>

<nature>org.eclipse.wst.common.modulecore.ModuleCoreNature</nature>

<nature>org.eclipse.jdt.core.javanature</nature>

<nature>org.eclipse.wst.common.project.facet.core.nature</nature>

<nature>org.eclipse.wst.jsdt.core.jsNature</nature>

</natures>

</projectDescription>

### 2. Create or replace srsw/.classpath

<?xml version="1.0" encoding="UTF-8"?>

<classpath>

<classpathentry kind="src" path="src"/>

<classpathentry combineaccessrules="false" kind="src" path="/srs">

<attributes>

<attribute name="org.eclipse.jdt.launching.CLASSPATH\_ATTR\_LIBRARY\_PATH\_ENTRY" value="srs"/>

</attributes>

</classpathentry>

<classpathentry combineaccessrules="false" kind="src" path="/srsi-shared"/>

<classpathentry combineaccessrules="false" kind="src" path="/util"/>

<classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.web.container"/>

<classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.module.container"/>

<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER"/>

<classpathentry combineaccessrules="false" kind="src" path="/srsa-shared"/>

<classpathentry kind="con" path="org.eclipse.jdt.junit.JUNIT\_CONTAINER/4"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/activemq.apache.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/commons.jakarta.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/ecs.jakarta.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/httpunit.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jasperreports.sourceforge.net"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jaxws.sun.com"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jboss.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/jfreechart.jchart.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/joda-time.joda.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/log4j.jakarta.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/rhino.mozilla.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/slf4j.org"/>

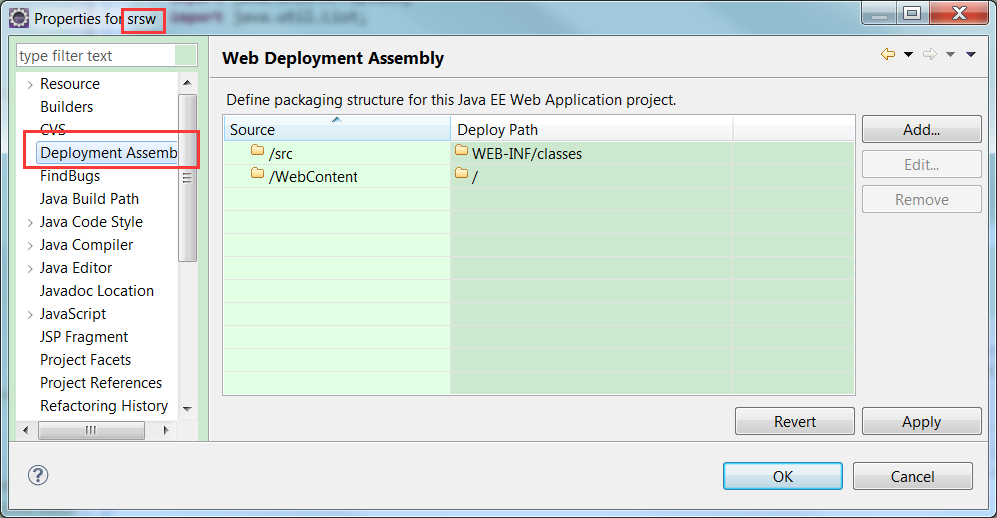
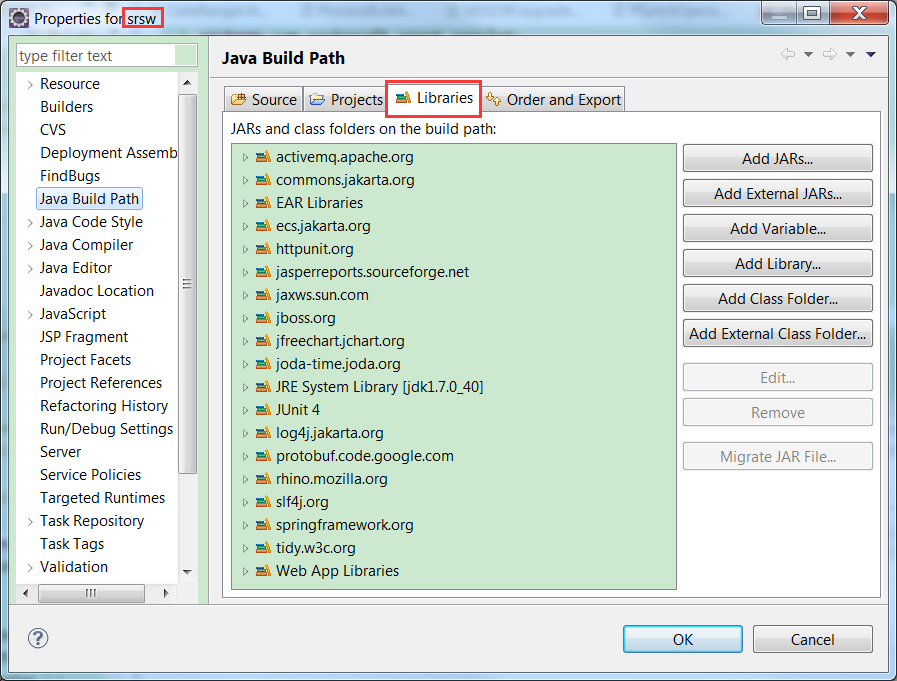
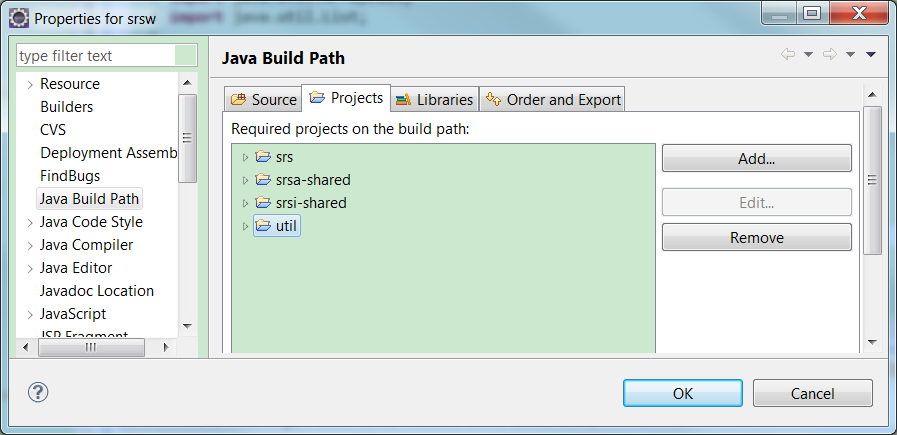
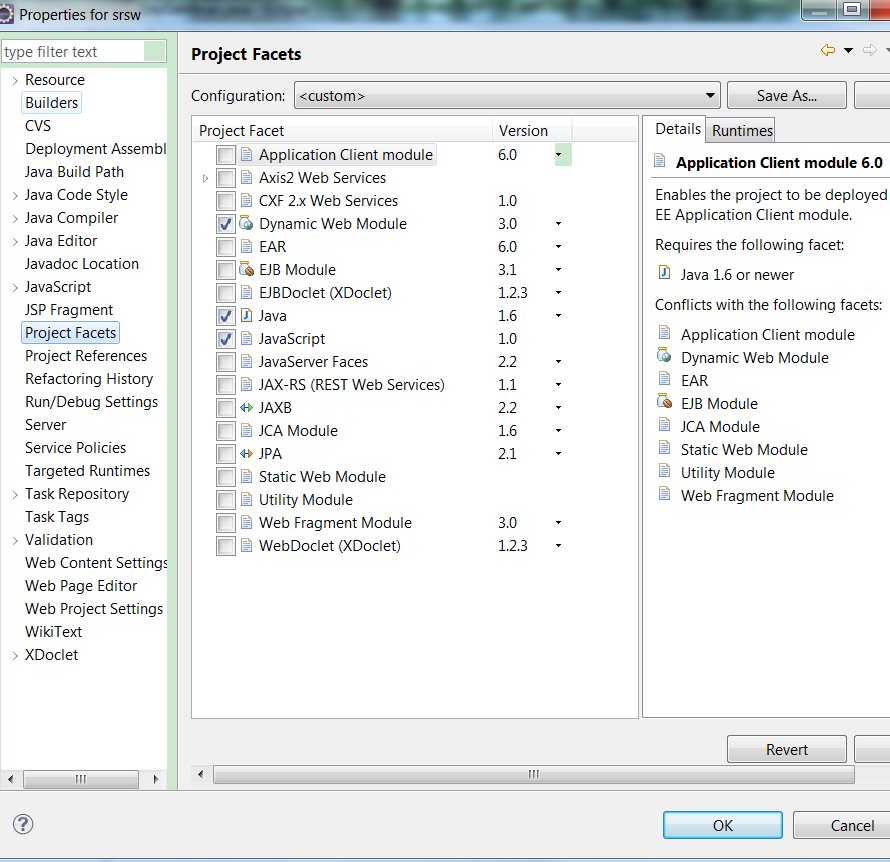
<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/springframework.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/tidy.w3c.org"/>

<classpathentry kind="con" path="org.eclipse.jdt.USER\_LIBRARY/protobuf.code.google.com"/>

<classpathentry kind="output" path="bin"/>

</classpath>



[Database upgrade utility](https://alm.aramark.eu/redmine/projects/3ed700ae-593c-4b62-aee0-14075dd53eb6/wiki/Database_upgrade_utility)

## Overview

DBUtil is a RealTime /FM support application for managing database upgrades from one version to the next. There are 3 elements to the application. These are:

1. Schema management
2. Migration scripts
3. Database upgrade utility

All elements are contained in the eclipse dbutil project which is located in cvs at location CVSROOT/srsproj/dbutil

## Schema management

All database changes are listed in /dbutil/webfiles/resources/schema.csv

Each row represents a single database schema operation. These are:

* CREATETABLE
* DEFINEKEY
* DROPTABLE
* ADDFIELD
* CHANGEFIELD
* DROPFIELD

The first field is the database version that the row refers to. For simplicity, the database version is now linked to the RealTime /FM release number. At time of writing the RealTime /FM release is 5.4.0 which translates to database version 540

The RealTime /FM database stores the current database version in installation.DB\_BUILD

The second field is the schema operation. The remaining columns provide arguments for the database operation.

Example row:  
  
540,ADDFIELD,actors,VARCHAR,50,,,,,LOGIN\_TOKEN

It is normal practice to group all rows relating to a table together.

**NOTE** Only open the schema.csv file in a text editor. Opening the file in Excel can render the file unusable for the dbutil application.

## Migration script

Each database version has an associated upgrade class. Upgrade classes are located in /dbutil/src/com/spokesoft/dbupgrade/

At time of writing the current upgrade class is Upgrade540.java

The upgrade class is used to perform data initialisation, migrations, adding new reports etc…

## Database upgrade utility

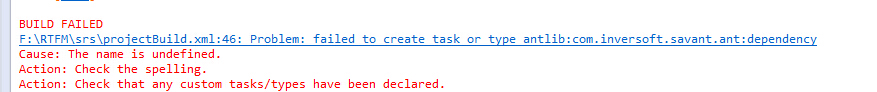
DBUtil is built as a enterprise archive (ear) and deployed to the JBoss deploy directory.

Once deployed, navigate to http://{jbosshost}:{jbossport}/dbutil/Dispatcher

The login details are:  
  
user: dba  
password: spokey

* Adjust the Database URL, Database User and Password to match your environment
* Enter the required database Build Number. Currently this is 540
* The default value for From Build Number is -1. The default behavior is to apply schema changes from the value in installation.DB\_BUILD to the value specified in Build Number. During the development cycle there may be situations where the developer needs to repeat a database upgrade. In this case installation.DB\_BUILD and Build Number will match and no upgrade will occur. Setting From Build Number to the (target build number -1) which in this instance is 539 will force the database upgrade utility to re-run all steps associated with the version number.

Build Exceptions:



It should print “[savant:dependency] Loaded artifact [spokesoft.com, targets, xdoclet-5.4.0.xml]”in normal.

Solution: A.Check if the “spk-savant-2.3.jar”locate in “ant\_home/lib” directory.

B.Check if the “Eclipse->Preferences->Ant->Runtime->Classpath->AntHome”view contains the ‘spk-savant-2.3.jar’;

2. java.lang.IllegalArgumentException: null object name org.jboss.mx.server.registry.BasicMBeanRegistry.get(BasicMBeanRegistry.java:509) org.jboss.mx.server.MBeanServerImpl.invoke(MBeanServerImpl.java:653) org.jboss.invocation.jrmp.server.JRMPInvoker$MBeanServerAction.invoke(JRMPInvoker.java:819) org.jboss.invocation.jrmp.server.JRMPInvoker.invoke(JRMPInvoker.java:420) org.jboss.invocation.jrmp.interfaces.JRMPInvokerProxy.invoke(JRMPInvokerProxy.java:133) org.jboss.invocation.InvokerInterceptor.invokeInvoker(InvokerInterceptor.java:365) org.jboss.invocation.InvokerInterceptor.invoke(InvokerInterceptor.java:197) org.jboss.proxy.TransactionInterceptor.invoke(TransactionInterceptor.java:61) org.jboss.proxy.SecurityInterceptor.invoke(SecurityInterceptor.java:70) org.jboss.proxy.ejb.HomeInterceptor.invoke(HomeInterceptor.java:184)

Check if there are duplicate file like “spk-srs\_xxx.ear”.

3.Taskman login error:”Invilad user”;Check if the actor have a device\_id.

4. Jasper Report run error:

“

2015-02-12 06:44:47,128 DEBUG [net.sf.jasperreports.engine.fill.SimpleTextLineWrapper] creating element font infos cache of size 2000

2015-02-12 06:44:47,130 ERROR [STDERR] net.sf.jasperreports.engine.util.JRFontNotFoundException: Font 'Arial' is not available to the JVM. See the Javadoc for more details.

2015-02-12 06:44:47,131 ERROR [STDERR] at net.sf.jasperreports.engine.fonts.FontUtil.checkAwtFont(FontUtil.java:344)

...”

Can’t find font. Add font jar package to jboss\lib or ejb.jar.

5.[jrcompile] File :

“C:\Users\jenkin.ge\workspace\srsr\src\com\spokesoft\jasper\TaskSignOff.jrxml ... FAILED.

[jrcompile] Error compiling report design : C:\Users\jenkin.ge\workspace\srsr\src\com\spokesoft\jasper\TaskSignOff.jrxml

[jrcompile] net.sf.jasperreports.engine.JRException: Errors were encountered when compiling report expressions class file:

[jrcompile] C:\Users\jenkin.ge\workspace\srs\TaskSignOff\_1443506085910\_599335.java:599: error: cannot find symbol

[jrcompile] value = (net.sf.jasperreports.engine.JRRenderable)(((com.spokesoft.reporting.util.SrsFunctions)variable\_SrsFunction.getValue()).getTaskQRCode(((java.lang.Integer)field\_TaskID.getValue()).intValue(),64,64)); //$JR\_EXPR\_ID=74$

[jrcompile] ^

[jrcompile] symbol: method getTaskQRCode(int,int,int)

[jrcompile] location: class SrsFunctions”

Solution: Delete extra jar in ‘$ant\_home/lib’.

6.”apt:

[apt] javac: directory not found: C:\Users\jenkin.ge\workspace\srsw\build

[apt] Usage: javac <options> <source files>

[apt] use -help for a list of possible options

[apt]

[apt] warning: The apt tool and its associated API are planned to be

[apt] removed in the next major JDK release. These features have been

[apt] superseded by javac and the standardized annotation processing API,

[apt] javax.annotation.processing and javax.lang.model. Users are

[apt] recommended to migrate to the annotation processing features of

[apt] javac; see the javac man page for more information.

[apt] warning: Annotation types without processors: javax.xml.bind.annotation.XmlRootElement,javax.xml.bind.annotation.XmlAccessorType,javax.xml.bind.annotation.XmlType,javax.xml.bind.annotation.XmlElement

[apt] 1 warning

[apt] Command invoked: apt -d C:\Users\jenkin.ge\workspace\srsw\build -s C:\Users\jenkin.ge\workspace\srsw\src -sourcepath C:\Users\jenkin.ge\workspace\srsw\src -g -Ar=..//srsw/build C:\Users\jenkin.ge\workspace\srsw\src\com\spokesoft\http\webservice\asp\OrgManagementHandler.java C:\Users\jenkin.ge\workspace\srsw\src\com\spokesoft\agent\servlet\SensorHandler.java C:\Users\jenkin.ge\workspace\srsw\src\com\spokesoft\agent\servlet\AgentEventHandler.java -classpath C:\Users\jenkin.ge\.savant\_repository\spokesoft.com\srs\spk-util-5.9.0.jar;C:\Users\jenkin.ge\.savant\_repository\spokesoft.com\srs\spk-srs-5.9.0.jar;C:\Users\jenkin.ge\.savant\_repository\spokesoft.com\srs\spk-httptest-5.9.0.jar;C:\Users\jenkin.ge\.savant\_repository\tomcat.apache.org\servlet\tomcat-servlet-2.4.jar;C:\Users\jenkin.ge\.savant\_repository\jboss.org\jboss\jboss-j2ee-4.0.4.jar;C:\Users\jenkin.ge\.savant\_repository\httpunit.org\httpunit\httpunit-1.6.jar;C:\Users\jenkin.ge\.savant\_repository\junit.org\junit\junit-3.8.1.jar;C:\Users\jenkin.ge\.savant\_repository\mozilla.org\rhino\js-1.6r1.jar;C:\Users\jenkin.ge\.savant\_repository\opensymphony.com\quartz\quartz-1.5.0-RC2.jar;C:\Users\jenkin.ge\.savant\_repository\jasperreports.sourceforge.net\jasperreports\jasperreports-5.2.0.jar;C:\Users\jenkin.ge\.savant\_repository\lowagie.com\itext\itext-2.1.7.jar;C:\Users\jenkin.ge\.savant\_repository\jchart.org\jfreechart\jfreechart-1.0.12.jar;C:\Users\jenkin.ge\.savant\_repository\jchart.org\jfreechart\jcommon-1.0.15.jar;C:\Users\jenkin.ge\.savant\_repository\jakarta.org\commons\commons-logging-1.1-dev.jar;C:\Users\jenkin.ge\.savant\_repository\jakarta.org\logging\log4j-1.2.13.jar;C:\Users\jenkin.ge\.savant\_repository\spokesoft.com\javax\spk-jaxws-all-08122006.jar;C:\Users\jenkin.ge\.savant\_repository\jboss.org\jboss\mail-4.0.4.jar;C:\Users\jenkin.ge\.savant\_repository\code.google.com\visualization\visualization-datasource-1.1.1.jar;C:\Users\jenkin.ge\.savant\_repository\code.google.com\guava\guava-17.0.jar;C:\Users\jenkin.ge\.savant\_repository\ibm.com\icu\icu4j-4.0.1.jar

BUILD FAILED

C:\Users\jenkin.ge\workspace\srs\projectBuild.xml:155: The following error occurred while executing this line:

C:\Users\jenkin.ge\workspace\srsw\build.xml:336: apt failed”

Solution: Create directory manually in ‘$srsw/build’