ellucian®

Banner Transformed: Working with Eclipse and Your Administrative Application

October 2017

ellucian

Ellucian
2003 Edmund Halley Drive
Reston, VA 20191
United States of America

ellucian

Table of Contents

ellucian

Introduction

This document includes information about how to configure a development environment for programming Banner Administrative Pages 9.x, also called “transformed” pages. This includes installing both the Eclipse IDE and Tomcat application server on a local computer as well as cloning, importing, and configuring the Banner source code.

Prerequisites

Before you proceed, you must successfully submit public keys to Ellucian for the Git repositories and receive credentials for the Ellucian Artifactory server. You access the Ellucian hosted Git repositories using SSH with the default port 22. You are then required to submit an RSA public key before you can get access.

- To generate and submit a public key to Ellucian, follow the instructions in the Ellucian Support Center *Article 9008: 1-1A3GVQO: Git repository RSA Key*.
- To request Artifactory credentials, follow the instructions in Ellucian Support Center *Article 34716: Requesting Ellucian Artifactory credentials for use in building Banner Administrative 9.0 source code*.

The following document has been updated for the Eclipse Plugins Foundations version 2.8.2, Frames 2.7.1 and Ellucian Templates 1.3.2.

Install Required Software

Each developer who will be developing Banner Administrative Pages, must install the following software on their workstation:

- Java Development Toolkit (JDK)
- Tomcat
- Eclipse Java EE IDE for Web Developers

Follow instructions provided by each vendor to install the software.

Platform

The following workstation platforms are acceptable for Eclipse and developing Banner 9 Administrative pages.

- Windows
- Mac OS
- Linux

Screen images in this document

All screen images in this document are from a Windows PC. There may be slight differences if your workstation is a Mac or you are running Linux.

Install JDK

JDK is an Oracle open source development environment.

1. Download JDK here:

   http://www.oracle.com/technetwork/java/javase/downloads.

2. Install JDK 7 or 8 based on the system specifications for your workstation.

   Here are links with instructions for installing the JDK:

   Linux:

   http://docs.oracle.com/javase/7/docs/webnotes/install/linux/linux-jdk.html#install-64-rpm

   Windows:

   http://docs.oracle.com/javase/7/docs/webnotes/install/windows/server-jre-installation-windows.html

3. Open a command window or terminal and validate the installation of your JDK using the following command:

   ```
   javac -version
   ```

Install Tomcat

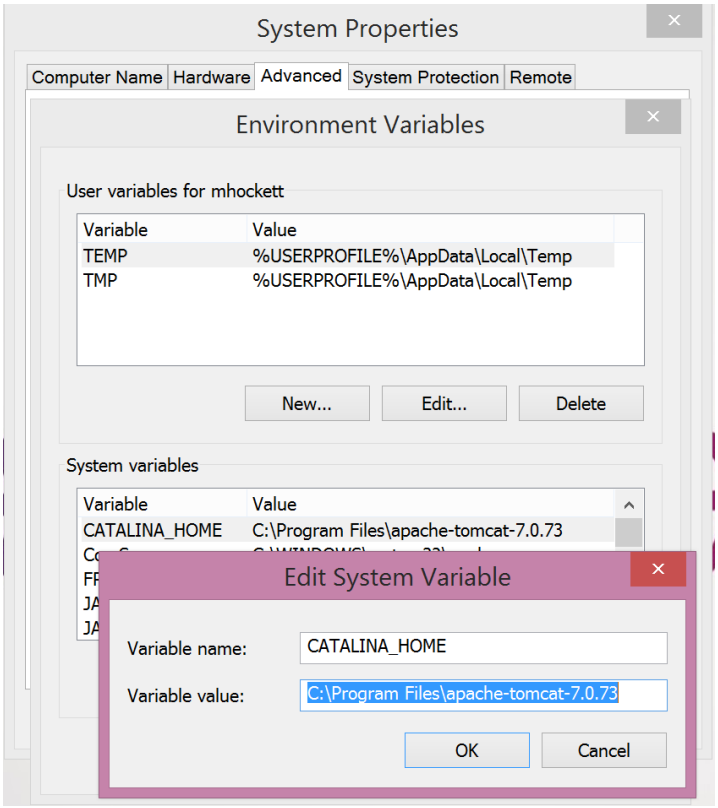
Tomcat is an Apache open source application server. Banner Administrative projects require Tomcat 7 or 8.

1. Download Tomcat v7.0 here:

   https://tomcat.apache.org/download-70.cgi

2. Follow the instructions on the Apache-Tomcat pages to install Tomcat.

3. Create the environment variable `CATALINA_HOME` in your system environment variables. The path is the location of your Tomcat installation.



Install Eclipse

You must install the Eclipse versions for Java EE IDE for Web Developers. Mars, Neon are currently supported. The Eclipse Morphis plugins are currently being tested using Oxygen. Luna is no longer supported.

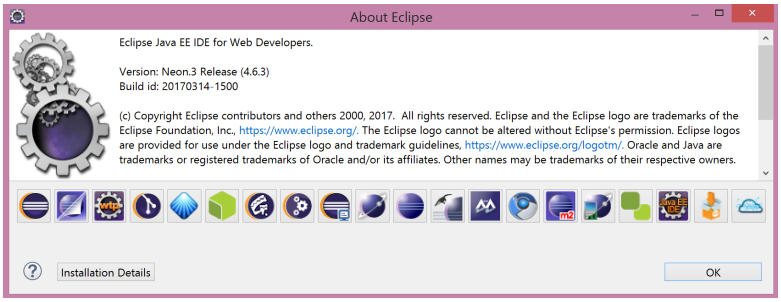
1. Download Eclipse here:

   http://www.eclipse.org/downloads.

2. Follow the instructions on the Eclipse download page to install Eclipse on your workstation.

3. After you have installed Eclipse, verify that you have installed Java EE for Web Developers by opening the application and the about Eclipse page. You may be prompted for a workspace. Choose the default workspace as you are only verifying the eclipse version.

You should see a window like the following.



Maven Settings for your Development Environment

Note: You must have your Ellucian Artifactory set up before setting up your environment.

To request Artifactory credentials, refer to the instructions available from the Ellucian Support Center in *Article 34716: Requesting Ellucian Artifactory credentials for use in building Banner Administrative 9.0 source code*.

The Banner Administrative Page Apps are built using Maven. You need to establish your Ellucian Artifactory ID and password in your maven settings for the Banner Admin Page build process.

1. Create an `.m2` directory under your user's home directory. The directory may already exist so check first to see if it does.

   ```
   mkdir c:/Users/<yourusername>/.m2
   ```

2. Copy the `settings.xml` file into this directory.

   You can download a sample `settings.xml` file from the Ellucian Support Center in *Article 000034790 - Maven settings.xml and pom.xml files for use with transformation source code*.

3. Edit the `settings.xml` file and change the `artifactoryuser` and `artifactorypassword` entries to your username and password.

Following is the code for a typical `settings.xml` file. You can use it as your initial `settings.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
 <settings xmlns="http://maven.apache.org/settings/1.0.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

   <!-- pluginGroups
    | This is a list of additional group identifiers that will be searched when resolving plugins
by their prefix, i.e.
    | when invoking a command line like "mvn prefix:goal". Maven will automatically add the group
identifiers
    | "org.apache.maven.plugins" and "org.codehaus.mojo" if these are not already contained in
the list.
    |-->
   <pluginGroups>
     <!-- pluginGroup
      | Specifies a further group identifier to use for plugin lookup.
     <pluginGroup>com.your.plugins</pluginGroup>
     -->

   <!-- jboss.org config start -->
     <pluginGroup>org.jboss.maven.plugins</pluginGroup>
```

```
   <!-- jboss.org config end -->

   </pluginGroups>

   <proxies>
   </proxies>

   <servers>
     <!-- server
      | Specifies the authentication information to use when connecting to a particular server,
identified by
      | a unique name within the system (referred to by the 'id' attribute below).
      |
      | NOTE: You should either specify username/password OR privateKey/passphrase, since these
pairings are
      |       used together.
      |
     <server>
       <id>deploymentRepo</id>
       <username>repouser</username>
       <password>repopwd</password>
     </server>
     -->
     <server>
       <id>internal</id>
       <username>your_artifactory_username</username>
       <password>your_artifactory_password</password>
        </server>
     <server>
       <id>snapshot</id>
       <username>your_artifactory_username</username>
       <password>your_artifactory_password</password>
     </server>
     <server>
       <id>ellucian</id>
       <username>your_artifactory_username</username>
       <password>your_artifactory_password</password>
     </server>
   </servers>

   <mirrors>
   </mirrors>

   <profiles>
   </profiles>

 </settings>
```

Bring up and configure Eclipse IDE

1. Create the following directory structure for your Administrative page development.

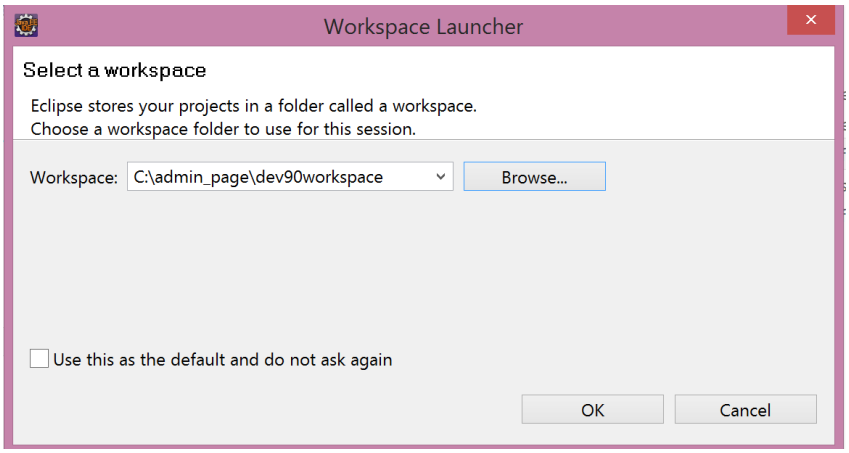
   ```
   mkdir c:/admin_page
   cd c:/admin_page
   mkdir dev90
   mkdir dev90workspace
   ```

You are going to clone the Banner Admin page GIT repository into `c:/admin_page/dev90` and use the `c:/admin_page/dev90workspace` as the Eclipse workspace.

Install Banner Admin Page Eclipse Plugins

You must install the *Foundations and Frames* plugin and the *Template* plugin in your Eclipse installation before you can work with the Banner Administrative Pages.

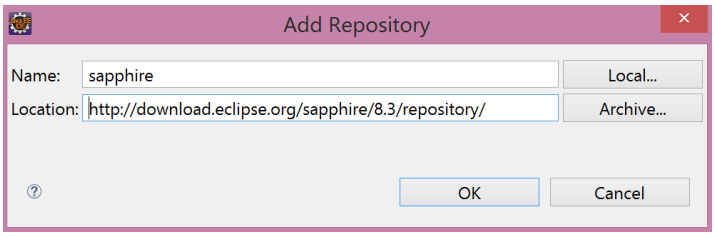
1. Open Eclipse.
2. In the Workspace window, browse to or enter `c:\admin_page\dev90workspace` as your project workspace.



Note: You must first have access to the Ellucian Artifactory to install the Banner Admin Page Eclipse Plugins.

3. Access **Help > Install New Software …** and click **Add** to add the plugin URL in Eclipse.
4. Install the third party Sapphire Eclipse plugin by entering a **Name** and the following URL in the **Location** for the plugin site.

   http://download.eclipse.org/sapphire/8.3/repository/



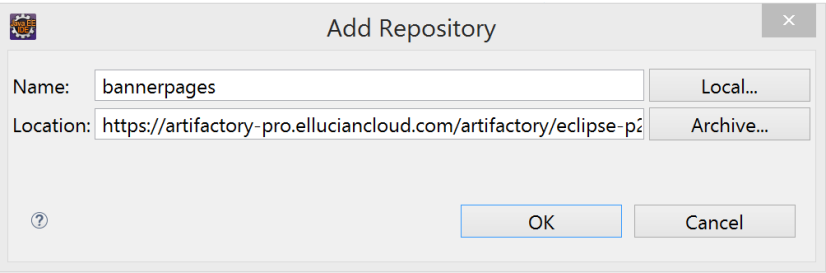
5. Click **OK** and select all on page to confirm the software installed.
6. Confirm license agreement.

Note: Do not restart eclipse until the next step to install the Foundations, Frames and Foundation Templates for Ellucian is finished.

7. Install the Foundations, Frames and Foundation Templates for Ellucian by entering a local **Name** and the following URL in the **Location** (URL) for the plugin site. You can choose any name, but the location must be one of the following URLs:

   - For a Mars, Neon and Oxygen versions of Eclipse, enter the URL:

     https://artifactory-pro.elluciancloud.com/artifactory/eclipse-p2-local/bannerpages-mars/



8. Click **OK**.
9. If prompted, enter your Ellucian Artifactory Username and Password.
10. If you are prompted for a “Secure Storage” entry, enter a **Password** and **Confirm password**.

    This master password entry enables Eclipse to keep your credentials in a key store so that you’re not required to enter them every time you update the Foundations and Frames plugin.
11. If prompted, enter security questions and answers that can be used for username and password recovery. Make note of the questions and answers.
12. In the Available Software window, click **Select All** to install all available plugins.
13. Click **Next** and accept the “Licenses” agreement.
14. Click **Finish**.
15. If prompted with a “Security Warning” about unsigned content, click **OK**.
16. After the install is complete, restart Eclipse. The restart is required.

Foundations Perspective

After the Foundations and Frames plugins are installed, you can view the project with the proper tools to perform Banner Admin Page development.

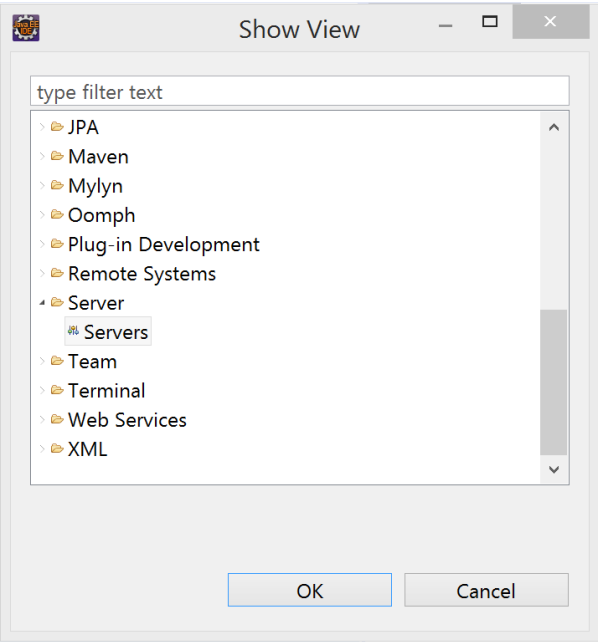
1. From the Window menu, select **Perspective > Open Perspective > Foundations**.
2. To restore the original Foundations perspective, select **Perspective** > **Reset Perspective**.

Add Server and Console Panels

Add the **Server** and **Console** panels.

1. Select **Windows > Show View > Servers** to show the server view.

If you do not see the Servers option here, then select **Windows > Show View > Other** and find Servers in the list.



2. Select **Windows > Show View > Other > General > Console** to show the console view.

Configure Eclipse for Foundations Framework

Open the Eclipse Preferences to modify the Foundation Settings specific to Ellucian Banner Admin Pages.

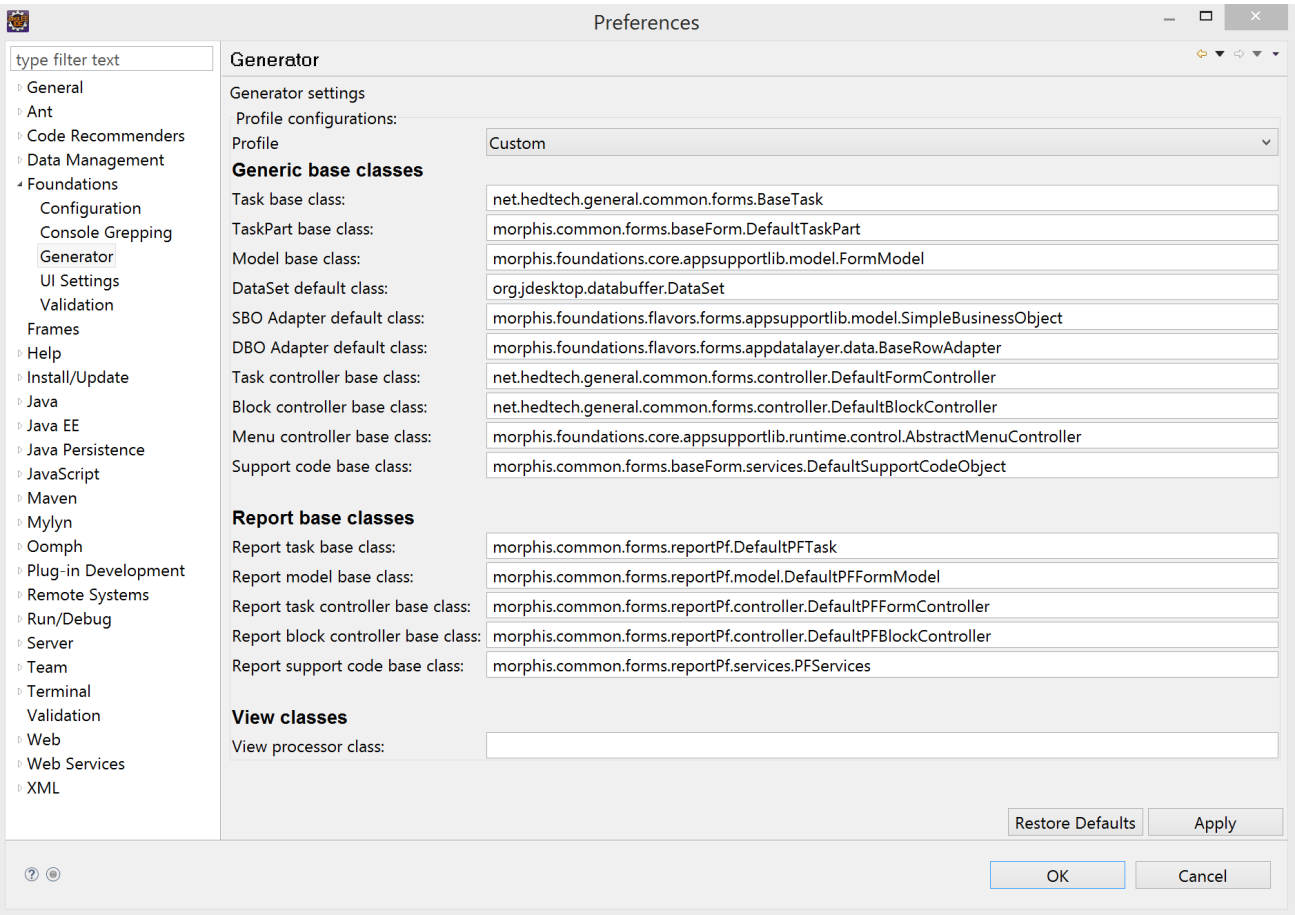
1. Add Foundations Generator Settings.

    1.1. Select **Windows > Preferences > Foundations Generator** settings.

    1.2. Select the **Generator tab** and the **Custom** Profile.

2. Enter values for all of the Generic base Classes and Report base classes as shown in the following image.

    Note: Older versions of the Foundations framework do not have rows for the Adapter and DataSet default classes. In addition, the word Form was changed to Task in the newer version. If you are missing these options, then use the **Help > Check for Updates** feature in Eclipse to update your Morphis plugins.

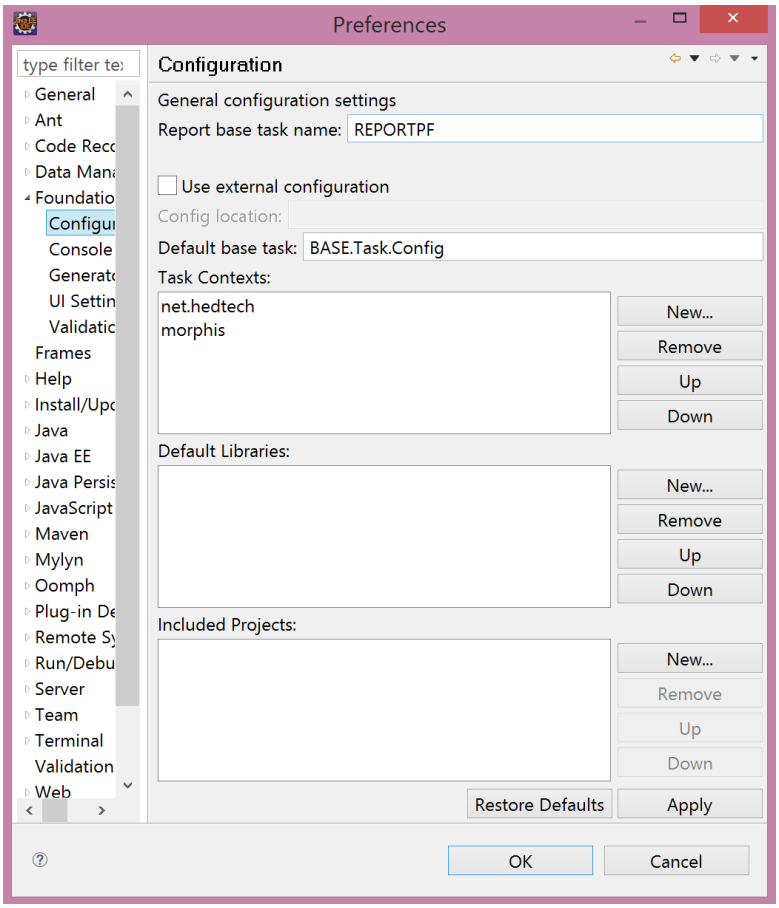


Copy and paste the class names from the following table.

Class	Class name
Task base class	net.hedtech.general.common.forms.BaseTask
TaskPart base class	morphis.common.forms.baseForm.DefaultTaskPart
Model base class	morphis.foundations.core.appsupportlib.model.FormModel
DataSet default class	org.jdesktop.databuffer.DataSet
SBO Adapter default class	morphis.foundations.flavors.forms.appsupportlib.model.SimpleBusinessObject
DBO Adapter default class	morphis.foundations.flavors.forms.appdatalayer.data.BaseRowAdapter
Task controller base class	net.hedtech.general.common.forms.controller.DefaultFormController
Block controller base class	net.hedtech.general.common.forms.controller.DefaultBlockController
Menu controller base class	morphis.foundations.core.appsupportlib.runtime.control.AbstractMenuController
Support code base class	morphis.common.forms.baseForm.services.DefaultSupportCodeObject
Report task base class	morphis.common.forms.reportPf.DefaultPFTask

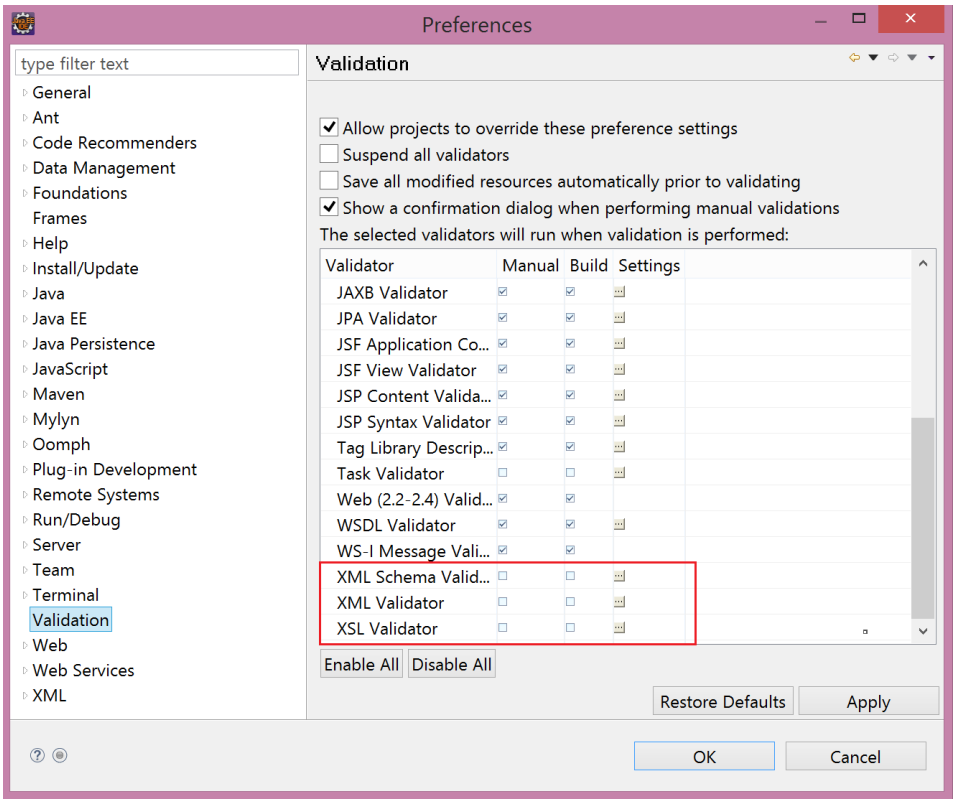
Class	Class name
Report model base class	morphis.common.forms.reportPf.model.DefaultPFFormModel
Report task controller base class	morphis.common.forms.reportPf.controller.DefaultPFFormController
Report block controller base class	morphis.common.forms.reportPf.controller.DefaultPFBlockController
Report support code base class	morphis.common.forms.reportPf.services.PFServices

3. Select **Window > Preferences > Foundations > Configuration**.
4. Add `net.hedtech` to the list of **Task Contexts**.
5. Change the **Default base task** to `BASE.Task.Config`.



6. Click **OK**.
7. Select **Window > Preferences >Validation**.

8. Uncheck both the Manual and Build options for the last three **Validators** (**XML Schema Validator**, **XML Validator**, and **XSL Validator**).



9. Click **Apply**.

   Eclipse will want to rebuild after you apply the above settings.

10. Click **OK**.

11. After the build finishes, exit Eclipse to prepare for cloning and importing the Ellucian repositories.

12. Eclipse allows you to bring all the settings described in this section from a file.

    12.1. Create a file morphis_config.ini with the following lines:

```
morphis.foundations.plugin.core/blockControllerBaseClass=net.hedtech.general.common.forms
.controller.DefaultBlockController
morphis.foundations.plugin.core/custom_defaultBlockControllerBaseClass=net.hedtech.genera
l.common.forms.controller.DefaultBlockController
morphis.foundations.plugin.core/custom_defaultDataSetClass=org.jdesktop.databuffer.DataSe
t
morphis.foundations.plugin.core/custom_defaultFormControllerBaseClass=net.hedtech.general
.common.forms.controller.DefaultFormController
morphis.foundations.plugin.core/custom_defaultMenuControllerBaseClass=morphis.foundations
.core.appsupportlib.runtime.control.AbstractMenuController
morphis.foundations.plugin.core/custom_defaultRowBaseClass=morphis.foundations.flavors.fo
rms.appdatalayer.data.BaseRowAdapter
morphis.foundations.plugin.core/custom_defaultSimpleBusinessObjectBaseClass=morphis.found
ations.flavors.forms.appsupportlib.model.SimpleBusinessObject
morphis.foundations.plugin.core/custom_defaultSupportCodeBaseClass=morphis.common.forms.b
aseForm.services.DefaultSupportCodeObject
morphis.foundations.plugin.core/custom_modelBaseClass=morphis.foundations.core.appsupport
lib.model.FormModel
```

```
morphis.foundations.plugin.core/custom_reportBlockControllerBaseClass=morphis.common.forms.reportPf.controller.DefaultPFBlockController
morphis.foundations.plugin.core/custom_reportFormControllerBaseClass=morphis.common.forms.reportPf.controller.DefaultPFFormController
morphis.foundations.plugin.core/custom_reportModelBaseClass=morphis.common.forms.reportPf.model.DefaultPFFormModel
morphis.foundations.plugin.core/custom_reportServicesBaseClass=morphis.common.forms.reportPf.services.PFServices
morphis.foundations.plugin.core/custom_reportTaskBaseClass=morphis.common.forms.reportPf.DefaultPFTask
morphis.foundations.plugin.core/custom_taskBaseClass=net.hedtech.general.common.forms.BaseTask
morphis.foundations.plugin.core/custom_taskPartBaseClass=morphis.common.forms.baseForm.DefaultTaskPart
morphis.foundations.plugin.core/defaultBaseTaskConfiguration=BASE.Task.Config
morphis.foundations.plugin.core/defaultRowBaseClass=morphis.foundations.flavors.forms.appdatalayer.data.BaseRowAdapter
morphis.foundations.plugin.core/defaultSimpleBusinessObjectBaseClass=morphis.foundations.flavors.forms.appsupportlib.model.SimpleBusinessObject
morphis.foundations.plugin.core/eclipse.preferences.version=1
morphis.foundations.plugin.core/formControllerBaseClass=net.hedtech.general.common.forms.controller.DefaultFormController
morphis.foundations.plugin.core/generatorProfile=Custom
morphis.foundations.plugin.core/menuControllerBaseClass=morphis.foundations.core.appsupportlib.runtime.control.AbstractMenuController
morphis.foundations.plugin.core/modelBaseClass=morphis.foundations.core.appsupportlib.model.FormModel
morphis.foundations.plugin.core/taskBaseClass=net.hedtech.general.common.forms.BaseTask
morphis.foundations.plugin.core/taskContexts=net.hedtech;morphis
org.eclipse.wst.validation/org.eclipse.wst.xsl.core.xsl=TF02
vals/org.eclipse.wst.xml.core.xml/global=FF03
vals/org.eclipse.wst.xsd.core.xsd/global=FF02162org.eclipse.wst.xsd.core.internal.validation.eclipse.Validator
vals/org.eclipse.wst.xsl.core.xsl/global=FF02
```

12.2. Locate the config.ini within eclipse/configuration folder and add this line to the bottom of the file.

```
eclipse.pluginCustomization=<path>/morphis_eclipse.ini
```

Clone Ellucian Banner Admin Page source repositories into Eclipse workspace

Before you clone the GIT repositories, you must submit a public and internal RSA key to Ellucian Customer Service to gain access to the repositories.

Note: These instructions assume you intend to clone the source code directly from the Ellucian GIT repository. If you will be using a local origin server instead, then work with your GIT Administrator to get the appropriate clone commands for your server.

1. Clone the Ellucian GIT repositories into your `C:/admin_page/dev90` directory.

   Each developer installation of Banner Admin Pages requires the following repositories.

   Note: You must update these repositories at the same time. If Ellucian delivers a new version, the base, build, common or workspace may all be impacted so you should pull all to ensure that you retrieve the required changes.

   ```
   cd c:/admin_page/dev90
   git clone ssh://git@source.ellucian.com/banner/pages/base
   git clone ssh://git@source.ellucian.com/banner/pages/build
   git clone ssh://git@source.ellucian.com/banner/pages/common
   git clone ssh://git@source.ellucian.com/banner/pages/reports
   git clone ssh://git@source.ellucian.com/banner/pages/workspace
   ```

2. Clone specific product Banner Admin Page repositories from the following repositories into `c:/admin_page/dev90`.

   ```
   cd c:/admin_page/dev90
   git clone ssh://git@source.ellucian.com/banner/pages/alumni
   git clone ssh://git@source.ellucian.com/banner/pages/arsys
   git clone ssh://git@source.ellucian.com/banner/pages/bdr
   git clone ssh://git@source.ellucian.com/banner/pages/extsol
   git clone ssh://git@source.ellucian.com/banner/pages/finaid
   git clone ssh://git@source.ellucian.com/banner/pages/finance
   git clone ssh://git@source.ellucian.com/banner/pages/general
   git clone ssh://git@source.ellucian.com/banner/pages/payroll
   git clone ssh://git@source.ellucian.com/banner/pages/positioncontrol
   git clone ssh://git@source.ellucian.com/banner/pages/student
   git clone ssh://git@source.ellucian.com/banner/pages/studentaid
   ```

3. You can verify the source of the repository with the following GIT command:

   ```
   cd c:/admin_page/dev90/student
   git remote -v
   ```

Update pom.xml for Ellucian Client Location

Update the `c:/admin_page/dev90/base/net.hedtech.banner/pom.xml` to replace the morphis-tech with the Ellucian Artifactory.

1. Open the `pom.xml` file in the following directory path:

   `c:/admin_page/dev90/base/net.hedtech.banner/pom.xml`

2. Change **both** of these repository names:

   `http://maven.morphis-tech.com/repository/internal/`

   `http://maven.morphis-tech.com/repository/snapshot/`

   to the following:

   `https://artifactory-pro.ellucian cloud.com/artifactory/morphis-release/`

   Note: You should change **both** original repositories to the same new repository name.

   You need to make this change in four places. The following sample shows the code sections highlighting the repository names that you need to change.

```
<distributionManagement>
    <snapshotRepository>
       <id>snapshot</id>
       <url>http://maven.morphis-tech.com/repository/snapshot/</url>
    </snapshotRepository>
    <repository>
       <id>internal</id>
       <url>http://maven.morphis-tech.com/repository/internal/</url>
    </repository>
 </distributionManagement>

 <repositories>
    <repository>
       <id>internal</id>
       <name>Archiva Managed Internal Repository</name>
       <url>http://maven.morphis-tech.com/repository/internal/</url>
    </repository>
    <repository>
       <id>snapshot</id>
       <name>Archiva Managed Internal Repository</name>
       <url>http://maven.morphis-tech.com/repository/snapshot/</url>
    </repository>
 <repository>
```

```
    <id>jrs-ce-releases</id>
    <name>JasperReports Server CE releases repository</name>
    <url>http://jaspersoft.artifactoryonline.com/jaspersoft/repo/</url>
  </repository>
</repositories>
```

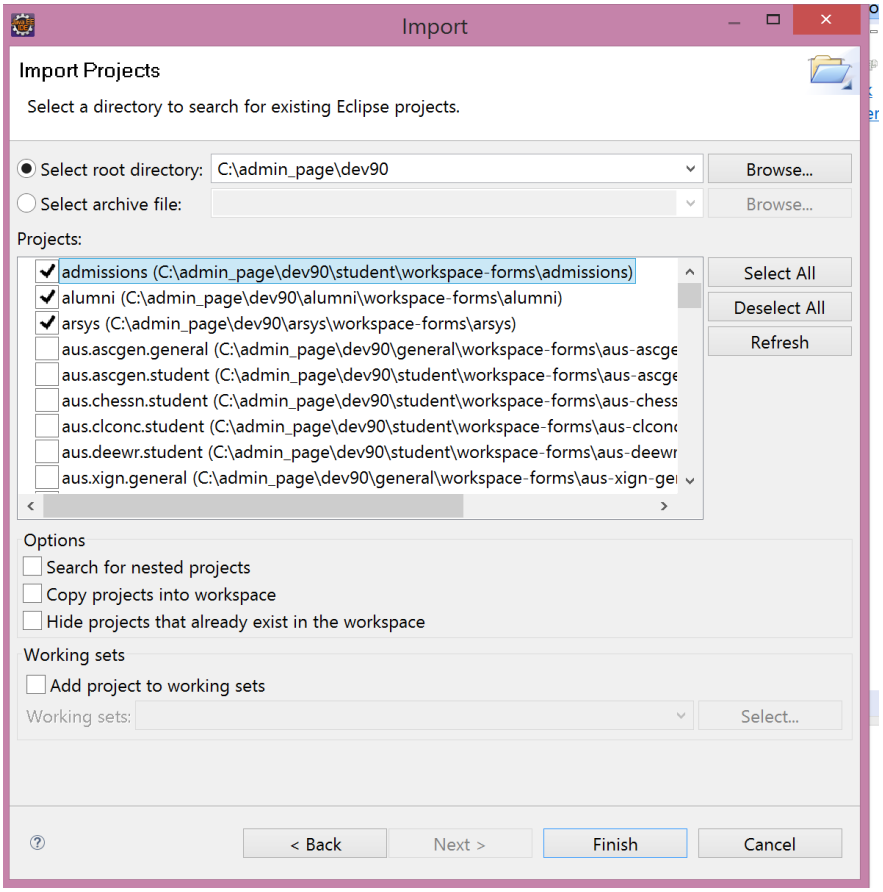
3. As noted in case 01631072 you may need to add the missing Jaspersoft repository. The missing repository is added with release 9.3.4. Add the following under the one for jrs-ce-releases.

```
<repository>
     <id>jrs-rest-java-client</id>
        <name>Jaspersoft clients releases</name>
     <url>http://jaspersoft.artifactoryonline.com/jaspersoft/jaspersoft-clients-releases/</url>
</repository>
```

Import Projects into Eclipse

1. Select **File > Import**.
2. Under General, select **Existing Projects into Workspace**.
3. Click **Next**.

4. Select the root directory of your cloned repositories `c:\admin_page\dev90`.



5. Click **Deselect All** to uncheck all projects. You may not want to import all projects because Eclipse will build everything you import and that will add extra overhead.

6. Select all the projects you want to import.

   At a minimum, you need to import the following projects:

   common-libraries

   document-management

   student-common

   general-common

   reports

   Webapp.HR.Workspace

   Webapp.HR.Services (from the build repository)

   Note: The Webapp workspace and services are named with “HR” not “Banner” because HR was the first application delivered in the series of transforming Banner products to Frames and Foundations. HR will soon be replaced with Banner.

   Ellucian recommends you import the Webapp.HR.Services from the build repository if you are planning to work on multiple projects, for example General, Student, Finaid and Arsys or Payroll, Position Control, Finance and General or plan on upgrading the common repos but not the

product repo. The Webapp HR Services will have the most current dependencies delivered with the common repos.

You may choose to import the Webapp.Student.Services from the Student repository or any of the others if you are only importing the projects listed in the dependencies.

Each product has its own Webapp services, which you can import in place of the Webapp.HR.Services from build if you are importing on that products projects. Note that the product service projects may not have the correct dependencies for Common version you install so it is highly recommended that the Webapp HR Service from build always be used.

7. The optional projects you can choose to import to support your app are the following:

   webapp.workspace.branding

8. The following table lists the product specific projects you can choose to import.

Source Product GIT Repository	**Projects You can Import**
General	general
	sevis.general
	bdr
	extsol
Alumni	alumni
Finaid	finaid
Finance	finance
Arsys	arsys
Student	admissions
	course
	degree
	learner
	student
	validation
Payroll	payroll
Positioncontrol	positioncontrol
Studentaid	studentaid

9. Select **Finish**.

   If you want to only work on Student projects, import the following projects:

   admissions

   course

   degree

   learner

   student

validation

If you want to only work on HR projects, import the following projects:

payroll

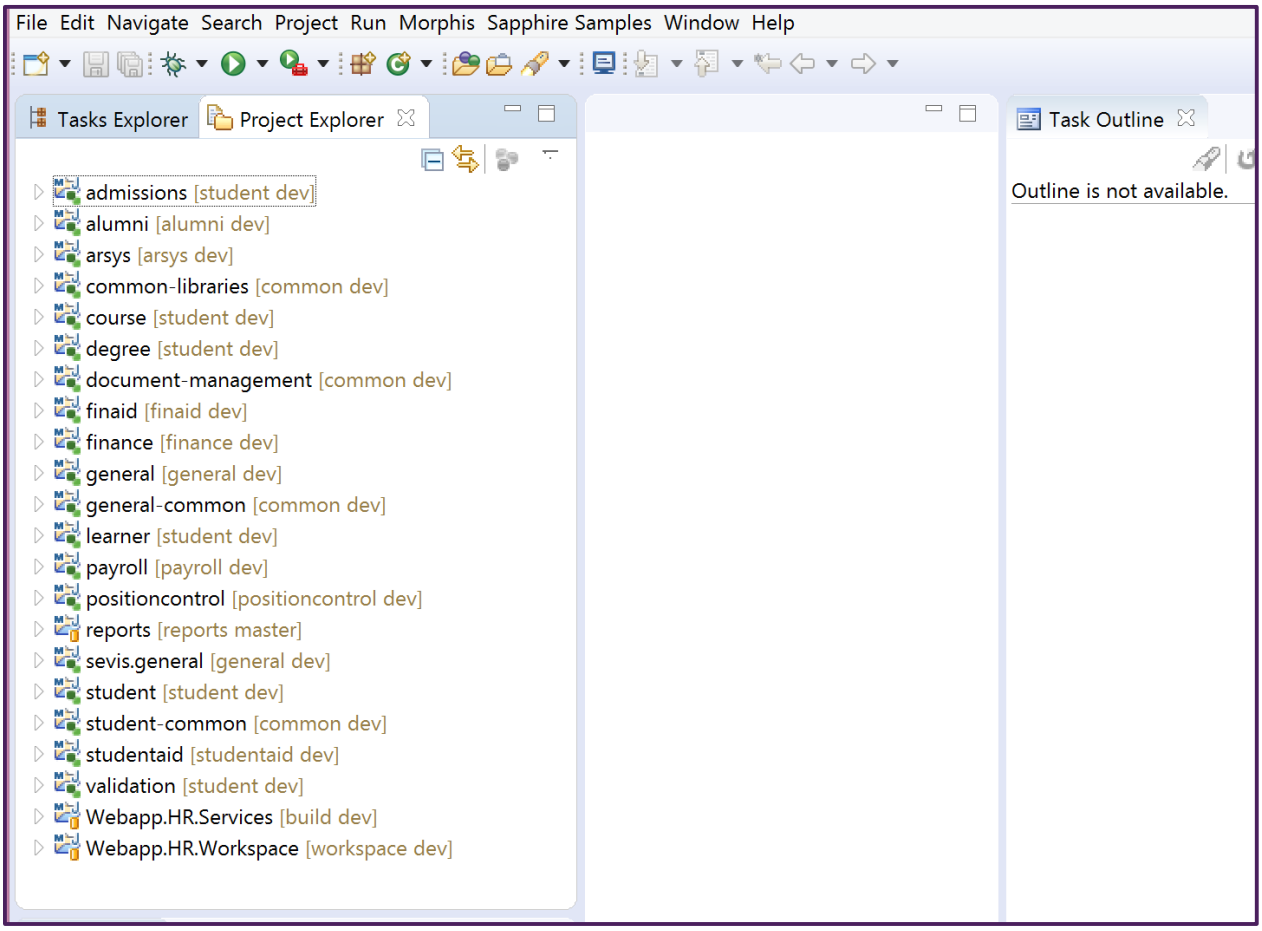
positioncontrol

Notes:

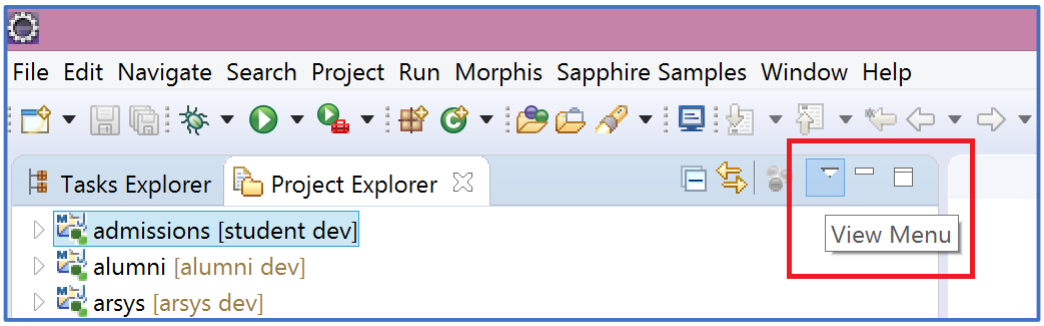
- Arsys is a swing product of Finance and Student; you may want to import it with both of those projects.
- Finaid may require Student and Arsys pages; you may want to import those projects if you are working on Finaid.
- Studentaid is international financial aid.
- Student-common is common to all products, not just the Student product. The pages in the student-common project are delivered to all products. The project would be more appropriately named banner-common.

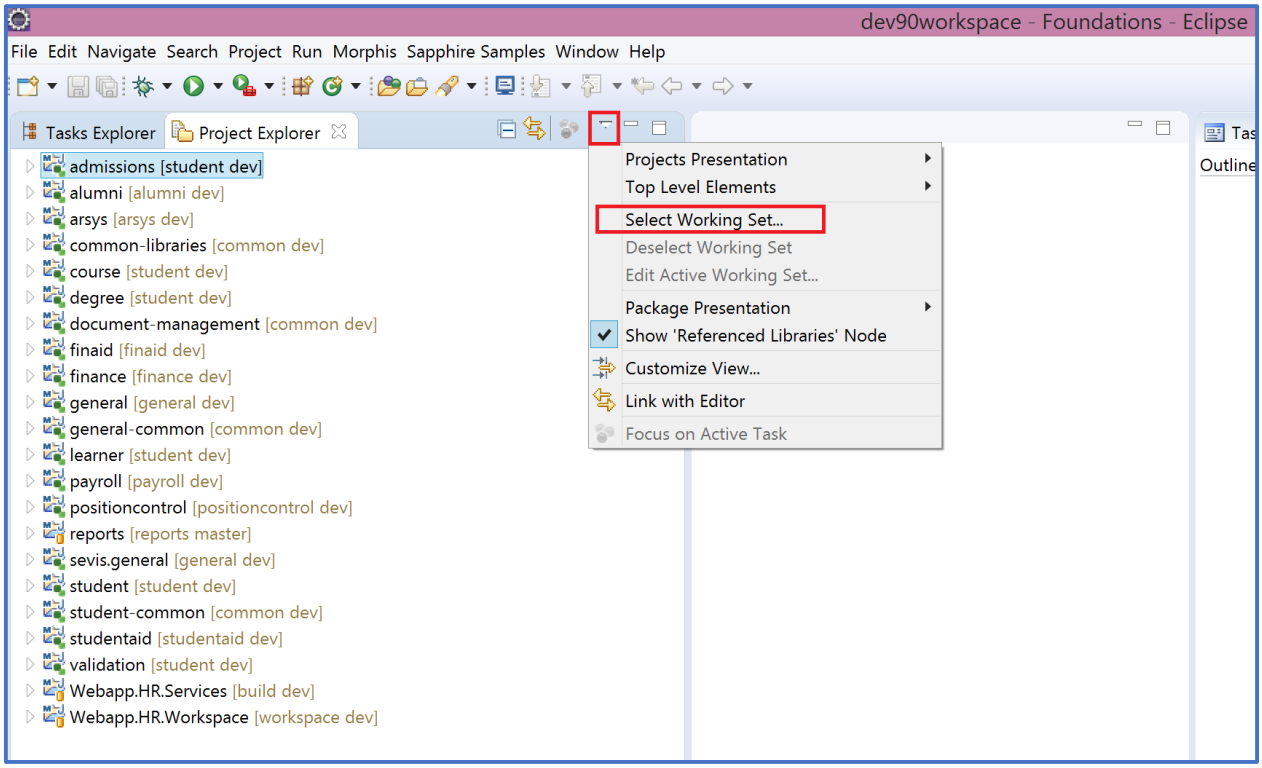
Your view will look like the following if you have selected all projects. Notice that the repository name and the branch are listed in parentheses beside each project.

Note: Finance-common was relocated to Finance as of the August 2017 Common 9.3.7 release. All General application pages located in the general-common project were relocated to the August 2017 General 9.3.5.



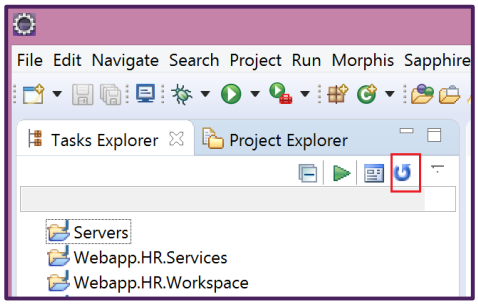
10. Select the **View Menu** (the solid downward arrow) and then select **Project Presentation > Hierarchical** to change the package presentation to Hierarchical.





11. Refreshing task explorer view

You may need to refresh the Tasks Explorer view after you upgrade your Foundations and Frames eclipse plugins . Click the refresh button to update the view.

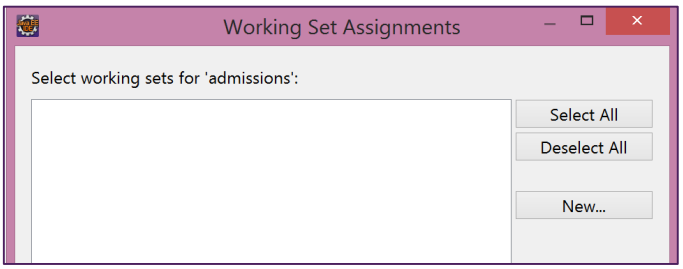


Optional Project Display options

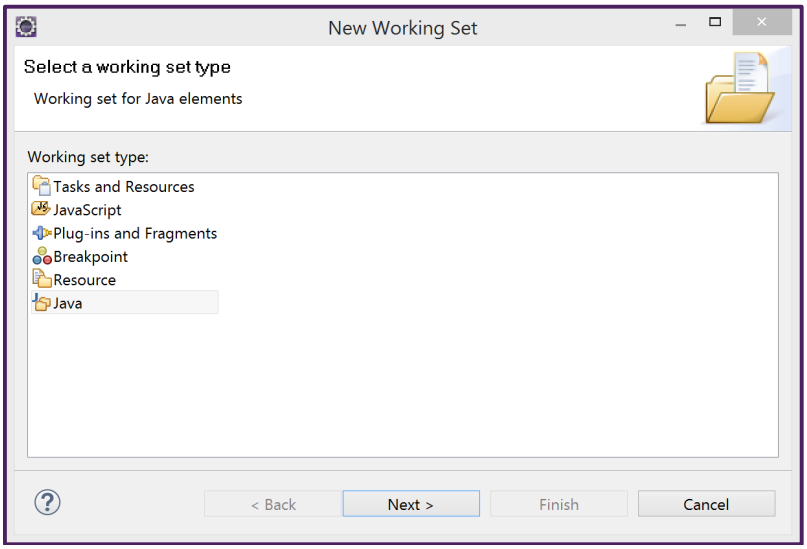
Eclipse allows creation of working sets to better organize the presentation of the projects.

1. Click on **Select Working Sets** from the **View Menu**.

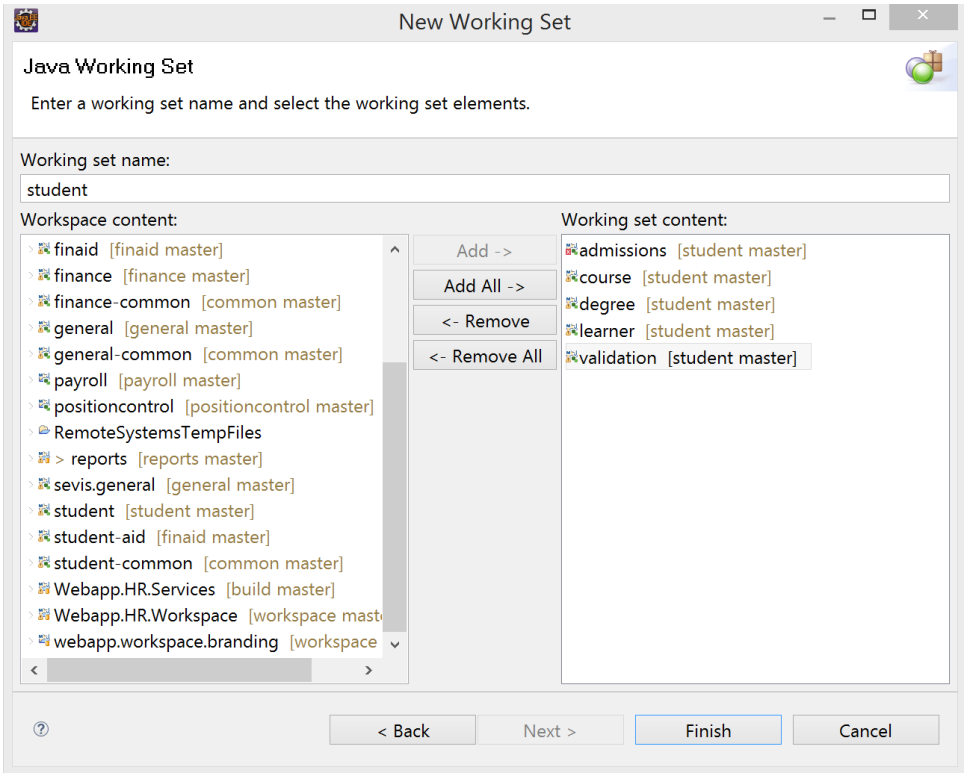
2. In the Working Set Assignments window, click **New**.



3. Select **Java** and click **Next**.

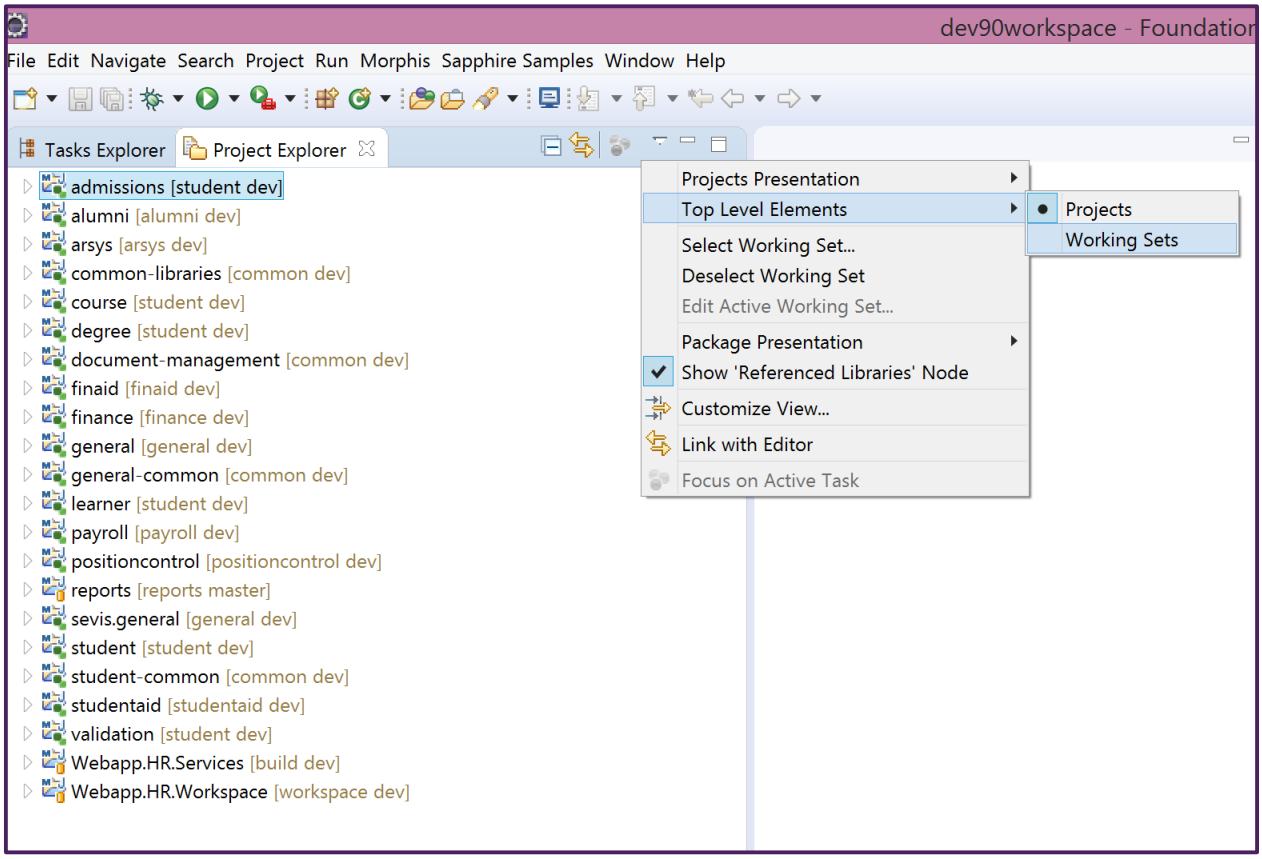


4. Enter a **Working set name** and select projects to be in the working set.

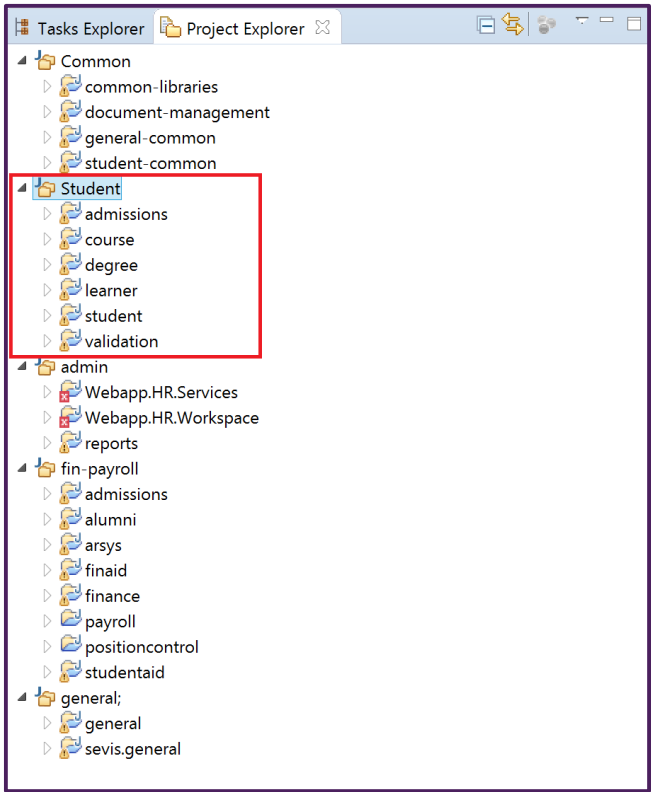


5. Click **Finish** when done.

6. Click the solid down arrow (View Menu) and select **Top Level Elements > Working Sets** to change the menu view to show working sets as the top level.



The result is that the view shows all the Student projects together.



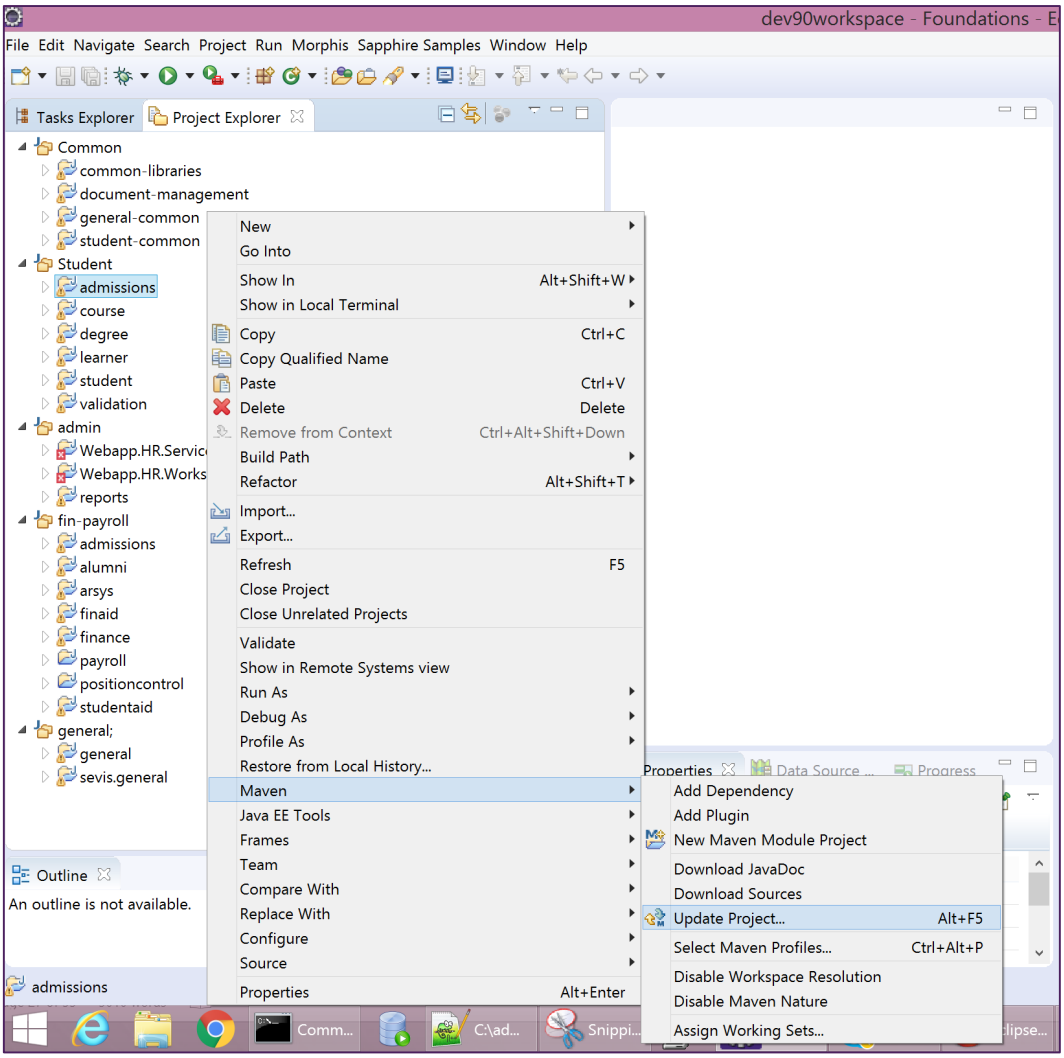
Configure Tomcat

The next step in setting up Eclipse for Banner Admin Page development is setting up Tomcat and the database server.

Update Maven

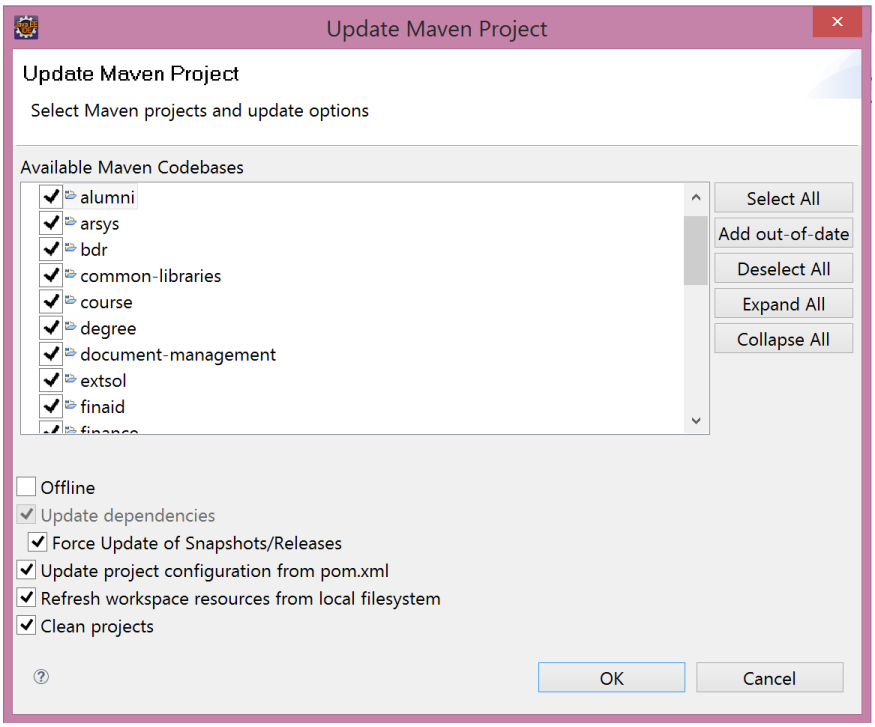
If you are setting up your environment for the first time, you need to run the Maven update process to bring in all repositories needed for Tomcat and the database server.

1. Right-click on one of your projects and select **Maven > Update Project**.



2. Click the **Select All** button to choose all projects.
3. Check the **Force Update of Snapshots/Releases** which will bring in current Maven repositories.

   The Maven repositories will be written to your `.m2/repositories` directory.



4. Click **OK**.

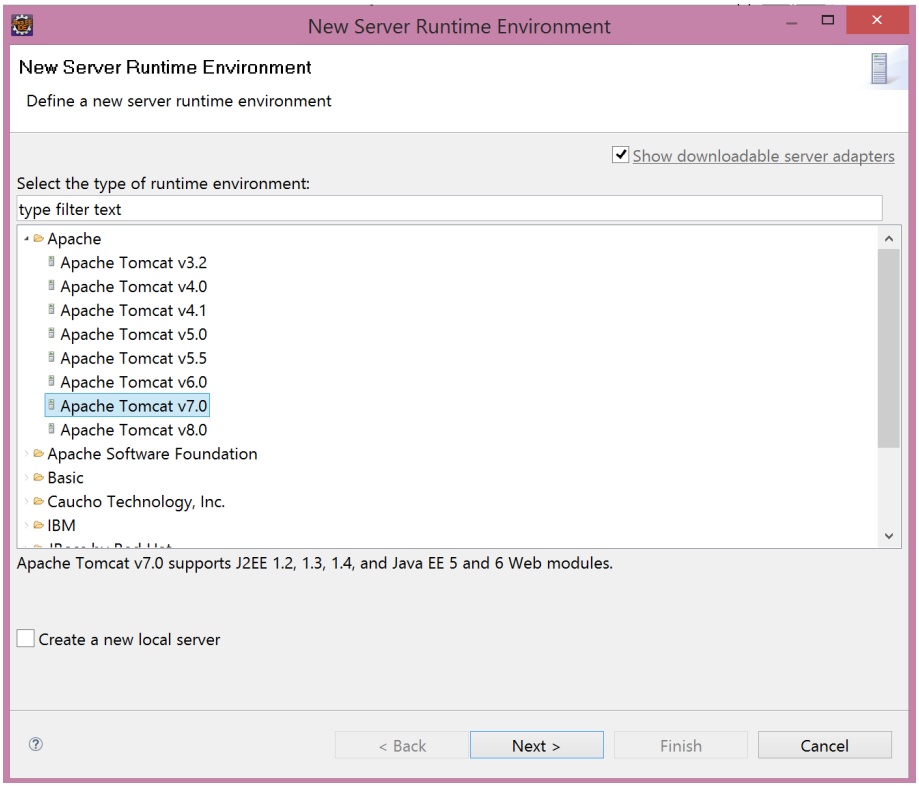
   The Maven update may take a while the first time you run it. After it is complete, you can proceed with setting up Tomcat and your Database Server.

Set up Tomcat

1. Open Eclipse and select **Window > Preferences.**
2. In the Preferences dialog box, navigate to **Server > Runtime Environment**.

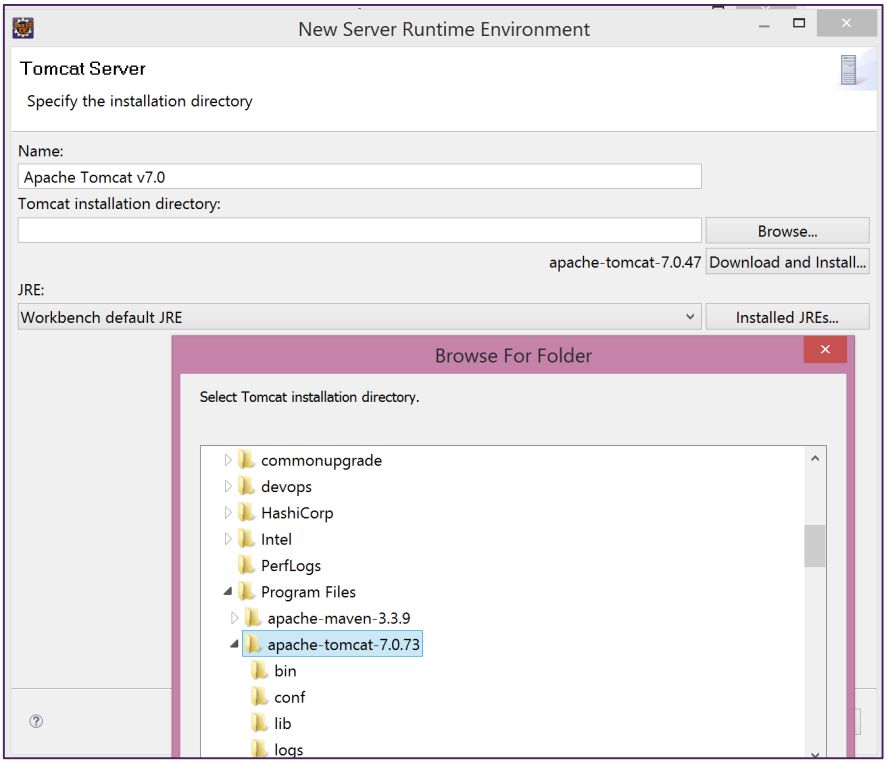
3. If the Apache Tomcat is not an available option, click the **Add** button to add the Apache Tomcat server.

    3.1. Under **Select the type of runtime environment**, select *Apache Tomcat v7.0* and click **Next**.



You will be prompted with the option to either download and install apache-tomcat-7.x.x server or select from where you installed it on your workstation.

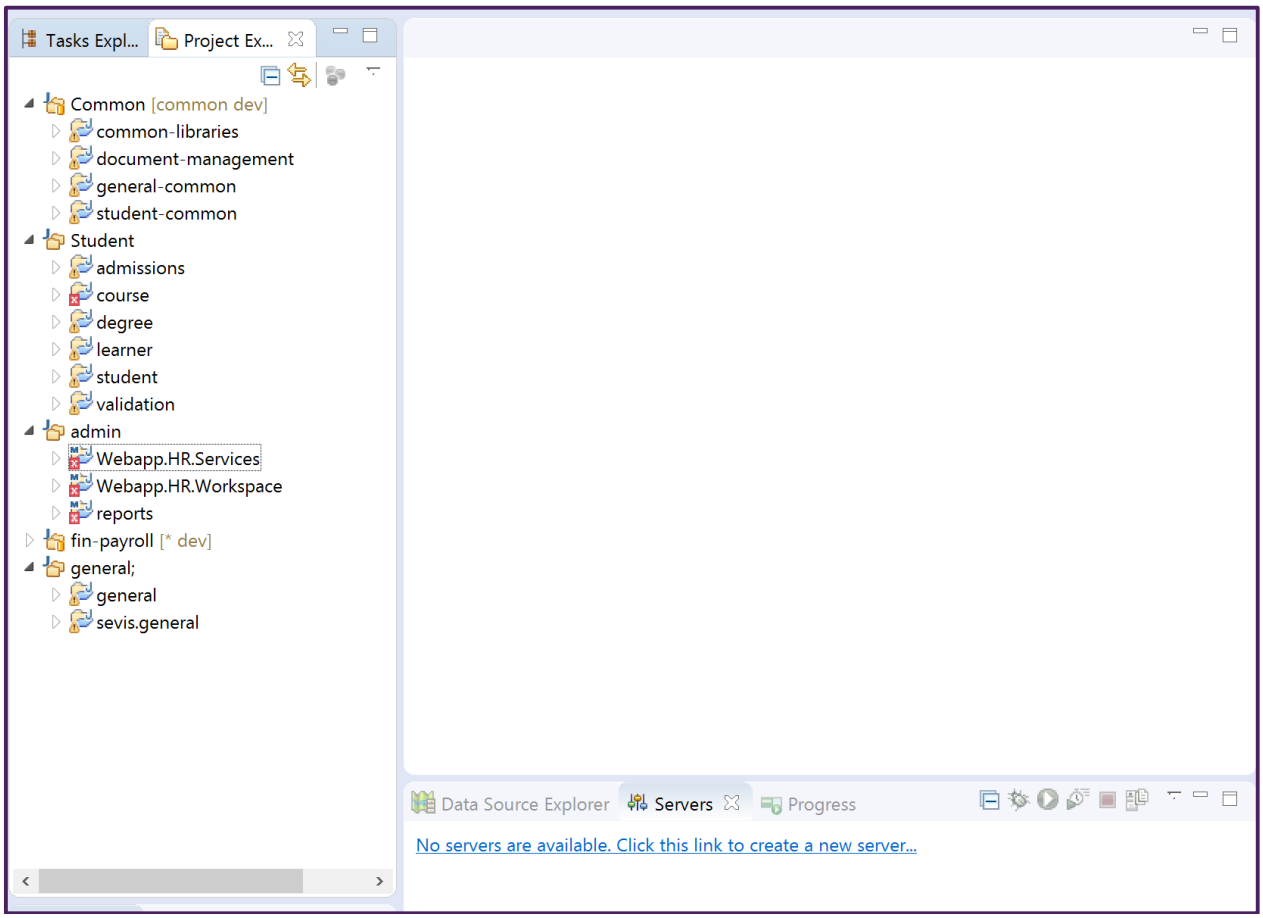
3.2. Browse to find the location where Tomcat is installed and select it.



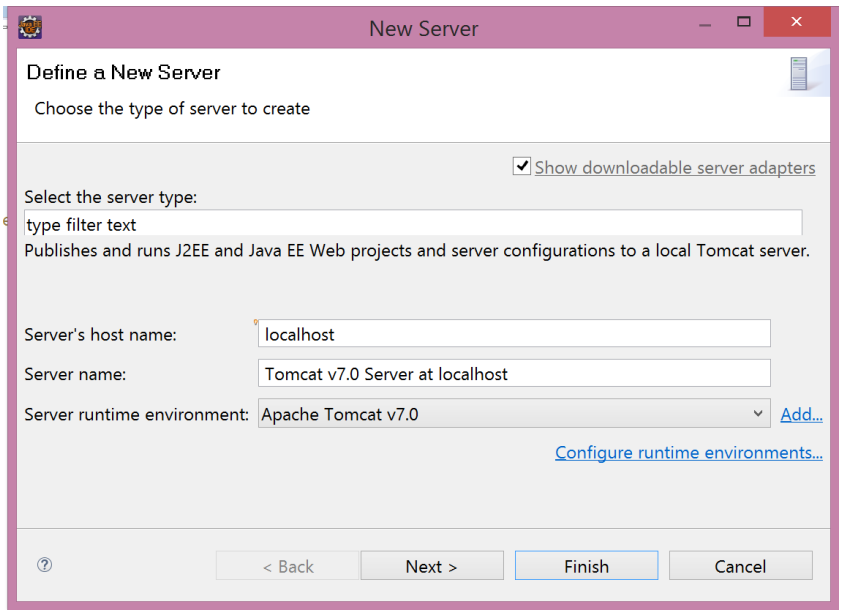
3.3. Click **Finish**.

4. Add Tomcat as the server for the project

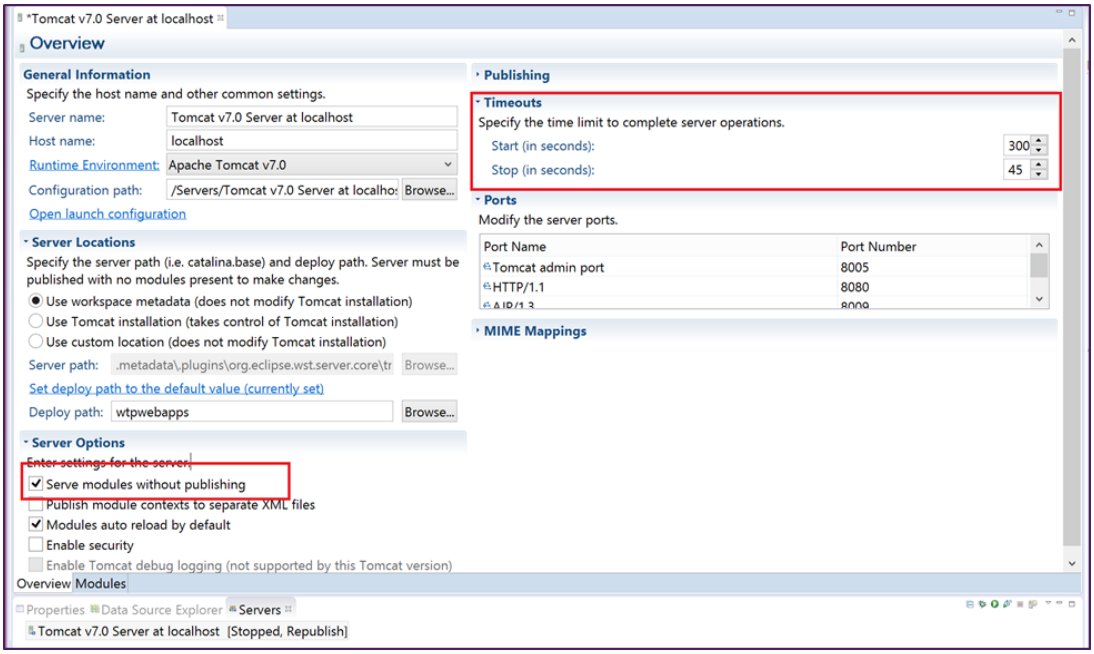
    4.1. On the **Servers** tab, click the **No servers are available** link to add Tomcat as the project server.



4.2. From the **Server runtime environment** field, select *Apache Tomcat 7.0* as the server and click **Finish**



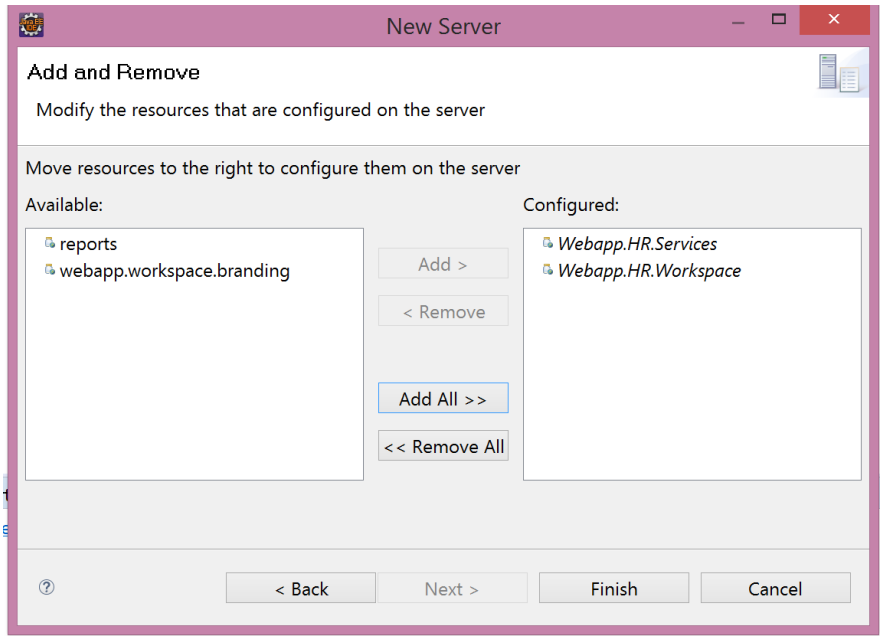
5. Click the Tomcat Server to open the properties and select the **Overview** tab.



5.1. Expand the **Timeouts** and increase the **Start** and **Stop**.

5.2. Check the **Serve modules without publishing** check box.

5.3. Select the **Modules** tab and add the Webapp.HR.Services and Webapp.HR.Workspace.

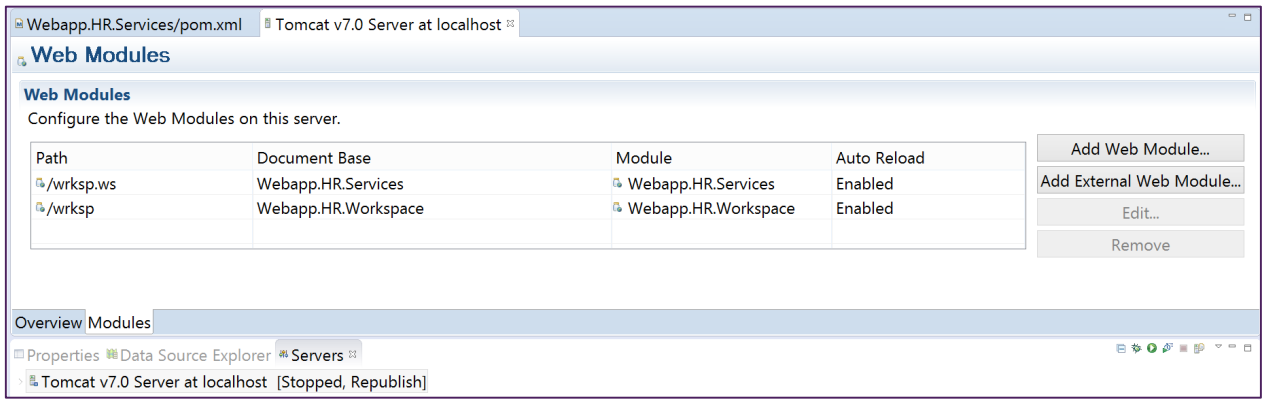


5.4. Click **Add Web Module** and select the Webapp.HR.Services.

If the path is not assigned, add it in `/wrksp.ws`.

5.5. Click **Add Web Module** again, select Webapp.HR.Workspace, and assign `/wrksp` to the path.

When finished, the Modules tab will look like the following.

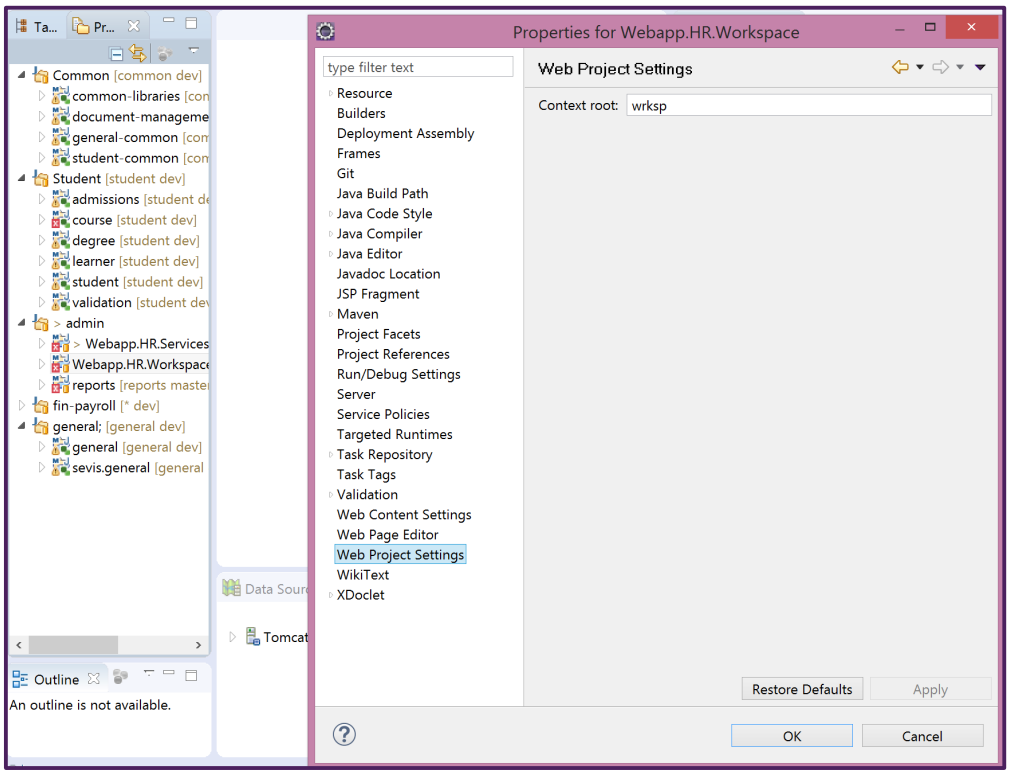


Note: In some instances the **Path** may revert back to the original value after submitting the change. If that happens, select the item and use the **Edit** button to make the change again.

6. Save the configuration.

You may want to verify the assignment as these two assignments are critical to running your app.

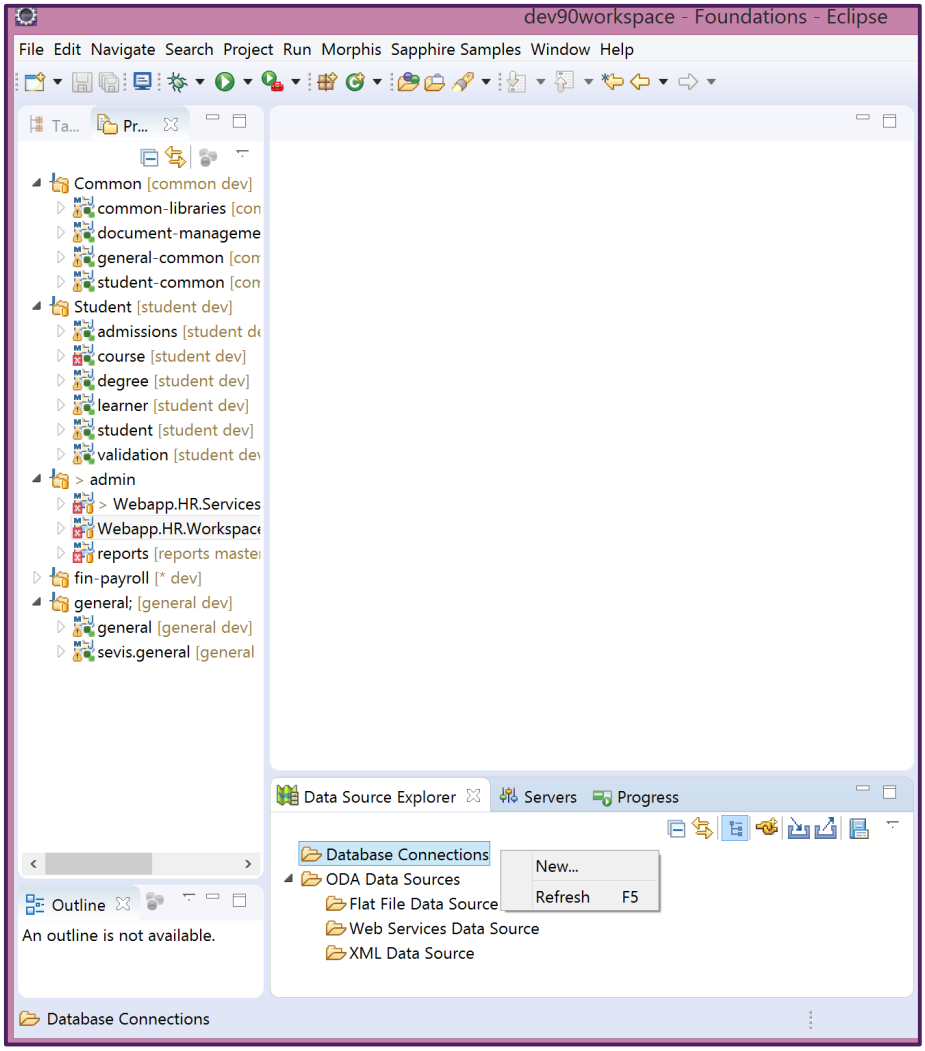
7. Right-click on the project and select **Properties > web project settings** to verify the context root as shown below.



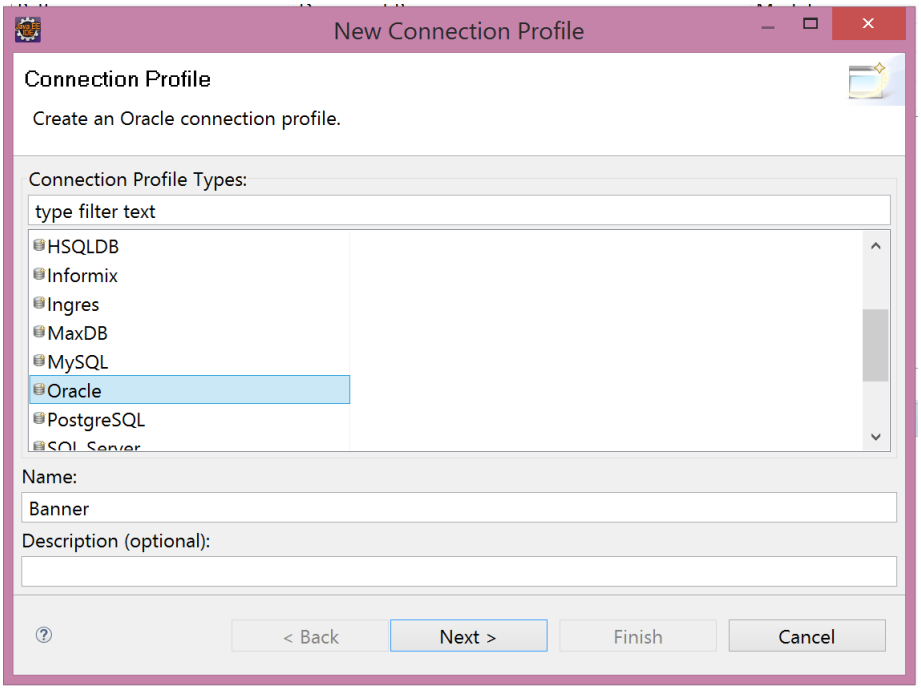
Database Configurations

The Foundations framework needs a database identified in the Data Source Explorer for table schema metadata.

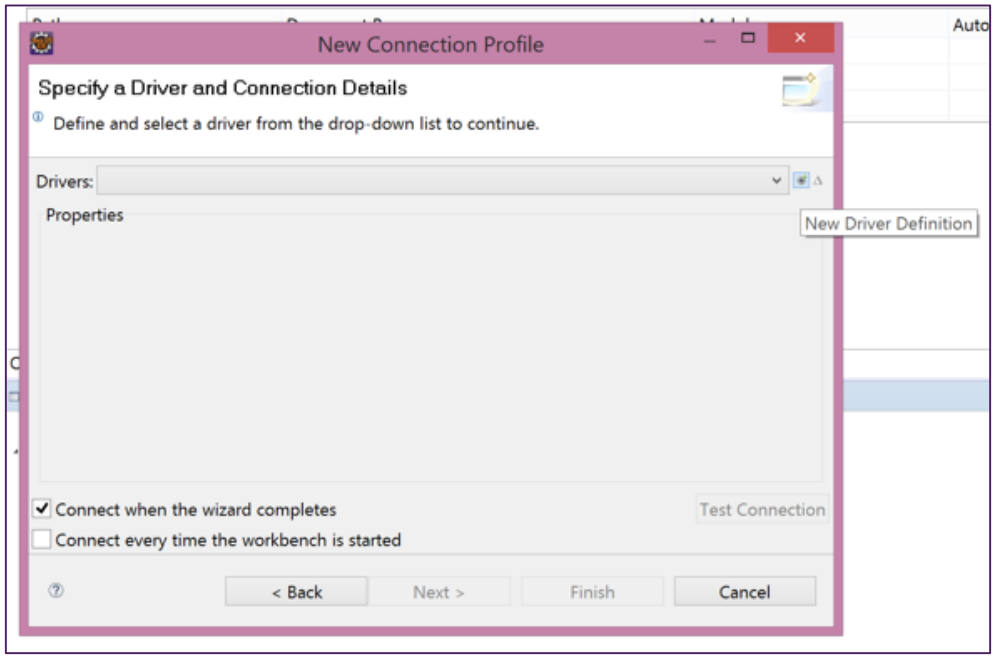
1. Select the **Data Source Explorer** tab.
2. Right-click on **Database Connections** and select **New**.



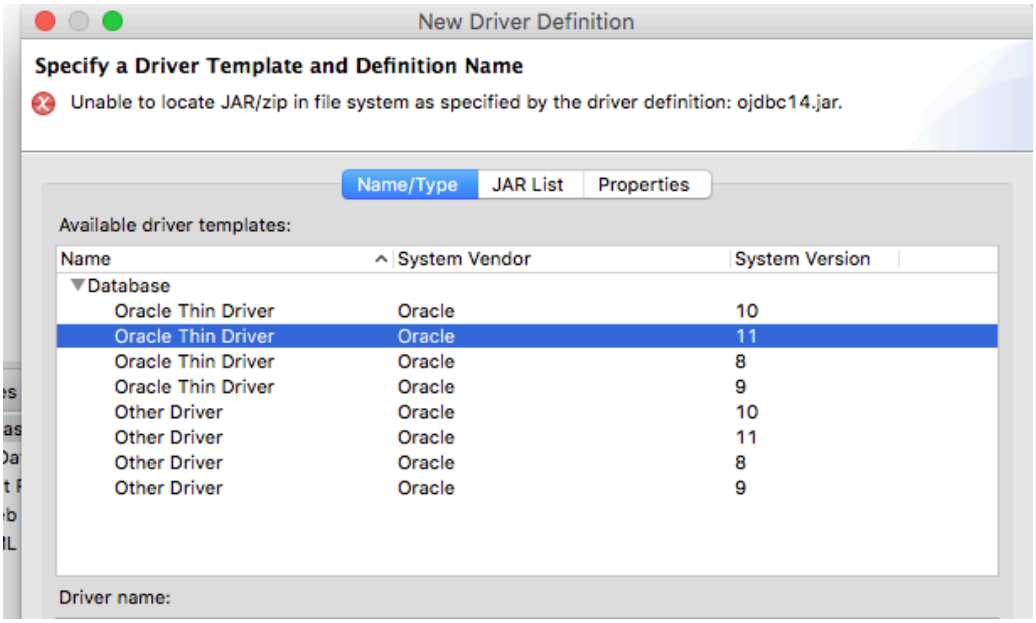
3. Under **Connection Profile Types**, select *Oracle* and enter a **Name**.



4. Click the **plus sign** next to the Drivers dropdown to add an Oracle driver.



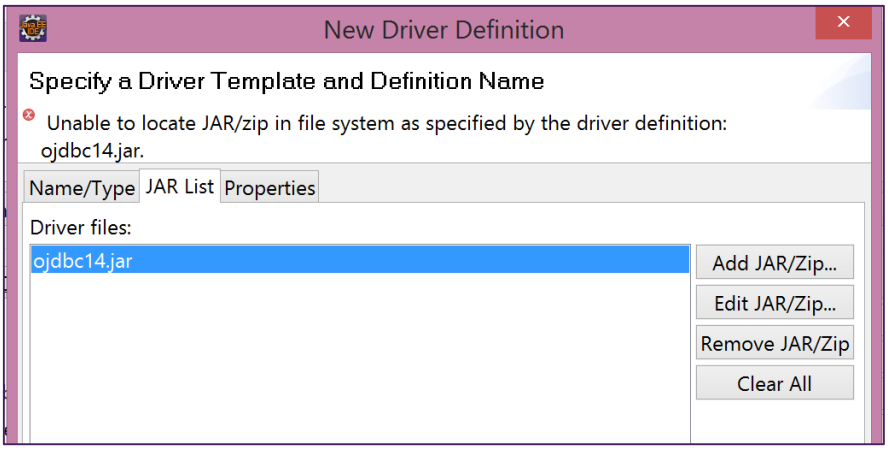
5. Select the **Oracle Thin Driver, Oracle 11 driver** template.



6. Select the **JAR List** tab.

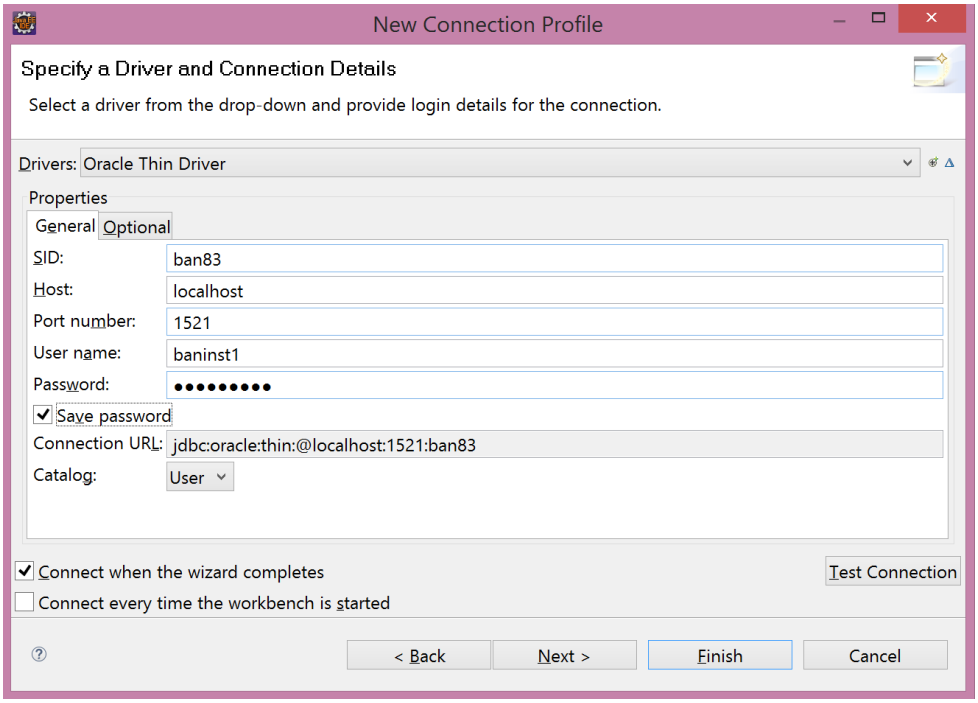
   6.1. Navigate to your `.m2/repositories/com/oracle` and select the `ojdbc jar` for the database version. If you do not have an `.m2/repositories/com/oracle` directory, you need to run the update Maven as described earlier in the Update Maven section.

   6.2. Remove the `ojdbc14.jar`.



7. Enter the SID, Host, User name and Password.

   This example shows a connection using a database installed locally on the same computer, however, you can also use a test database server.



8. You can click the **Test Connection** button to confirm that the database connection information is valid.

9. Click **Finish**.

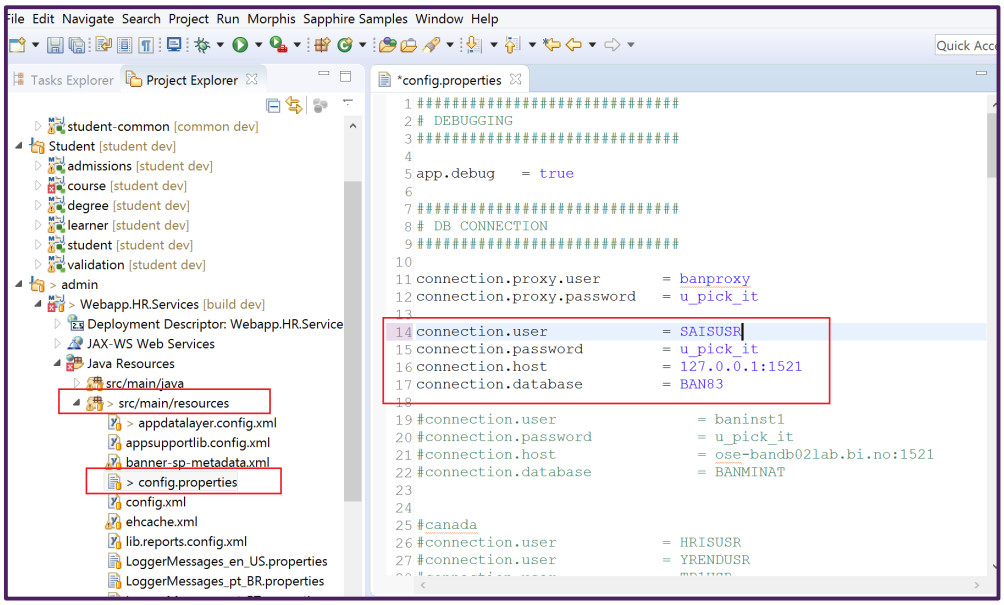
The setup process is complete.

Project Configurations

1. Navigate to the Webapp.HR.Services project.

Application specific configuration files are in the `src/main/resources` folder.

2. Modify the **config.properties** file to reflect your database information. This will be necessary for direct login to the database to succeed.



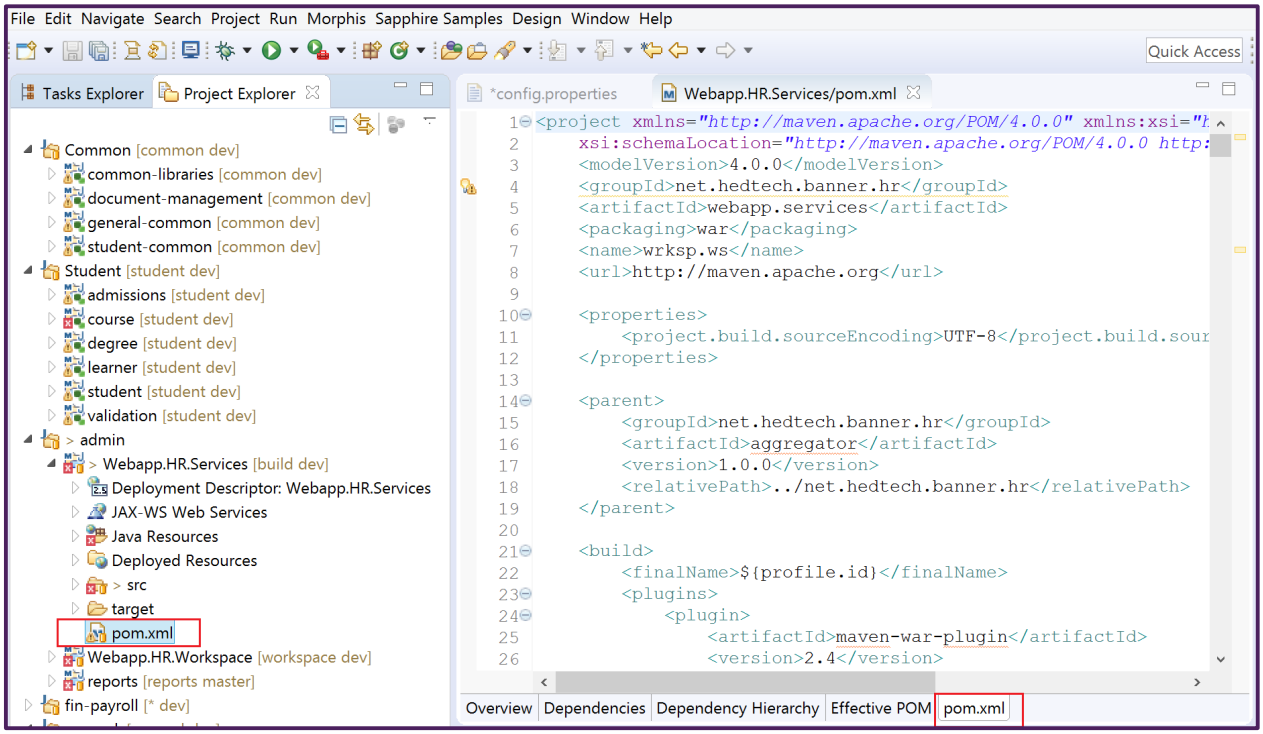
3. Modify `Appdatalayer.config.xml` by uncommenting the line where you map PKG_XML to the goktxml package. This setting is usually around line 52 of the file.

```
<property name="envVars">
    <map>
        <entry key="PKG_XML" value="goktxml"/>
        <entry key="useDBInstanceNLSSettings" value="${env.useDBInstanceNLSSettings}"/>
    </map>
</property>
```

Workspace Services pom.xml

You need to set up the project dependencies in the workspace services `pom.xml` file before you can build the project.

1. Locate the pom.xml in the workspace services as shown in the following image, and then click the **pom.xml** tab to view the source.



All Banner projects are listed in the dependency group.

2. You need to remove and retain just the projects that you have imported into your project. You must also retain the required projects from common.

   Warning: If you remove a project that you have imported, you may get build errors and you will not be able to run any of the pages from that project.

   The following are ***required*** dependencies and must stay in the pom.xml file. Note that if your site does not license Finance you are able to remove that dependency.

```
<!—must keep common-libraries, document-management, general-commmon,
                  student-common, finance-common -->
<dependency>
        <groupId>net.hedtech.banner.common</groupId>
        <artifactId>common-libraries</artifactId>
        <version>${project.parent.version}</version>
 </dependency>
 <dependency>
        <groupId>net.hedtech.banner.common</groupId>
        <artifactId>document-management</artifactId>
        <version>${project.parent.version}</version>
 </dependency>
 <dependency>
        <groupId>net.hedtech.banner.common</groupId>
        <artifactId>general-common</artifactId>
        <version>${project.parent.version}</version>
 </dependency>
 <dependency>
        <groupId>net.hedtech.banner.common</groupId>
        <artifactId>student-common</artifactId>
        <version>${project.parent.version}</version>
 </dependency>
```

The following are ***optional*** and are included in the Webapp.HR.Services from build `pom.xml` file. You need to review the dependencies in the services you import to verify that all projects you imported are accounted for.

To comment out a dependency, use the HTML comment tags <!-- followed by -- >.

```
<!-- comment out or remove projects not imported -->
  <dependency>
        <groupId>net.hedtech.banner.general</groupId>
        <artifactId>general</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.finance</groupId>
        <artifactId>finance</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.general</groupId>
        <artifactId>sevis.general</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.alumni</groupId>
        <artifactId>alumni</artifactId>
        <version>${project.parent.version}</version>
  </dependency>
  <dependency>
        <groupId>net.hedtech.banner.arsys</groupId>
        <artifactId>arsys</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.bdr</groupId>
        <artifactId>bdr</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.extsol</groupId>
        <artifactId>extsol</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.positioncontrol</groupId>
        <artifactId>positioncontrol</artifactId>
        <version>${project.parent.version}</version>
  </dependency>

  <dependency>
        <groupId>net.hedtech.banner.payroll</groupId>
        <artifactId>payroll</artifactId>
        <version>${project.parent.version}</version>
  </dependency>
```

```
<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>admissions</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>course</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>degree</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>learner</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>validation</artifactId>
        <version>${project.parent.version}</version>
</dependency>
<dependency>
        <groupId>net.hedtech.banner.student</groupId>
        <artifactId>student</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.finaid</groupId>
        <artifactId>finaid</artifactId>
        <version>${project.parent.version}</version>
</dependency>

<dependency>
        <groupId>net.hedtech.banner.studentaid</groupId>
        <artifactId>student-aid</artifactId>
        <version>${project.parent.version}</version>
</dependency>
```

Ojdbc, Ucp Jar files and Tomcat

Copy the ojdbc and ucp jar files from the .m2/repositories/com/oracle/ojdbc and ucp directories to the tomcat/lib folder to resolve the two com.oracle dependencies in the pom.xml.

Here are the dependencies in the Workspace Service pom.xm file that will be resolved once the two files are copied to the tomcat lib folder.

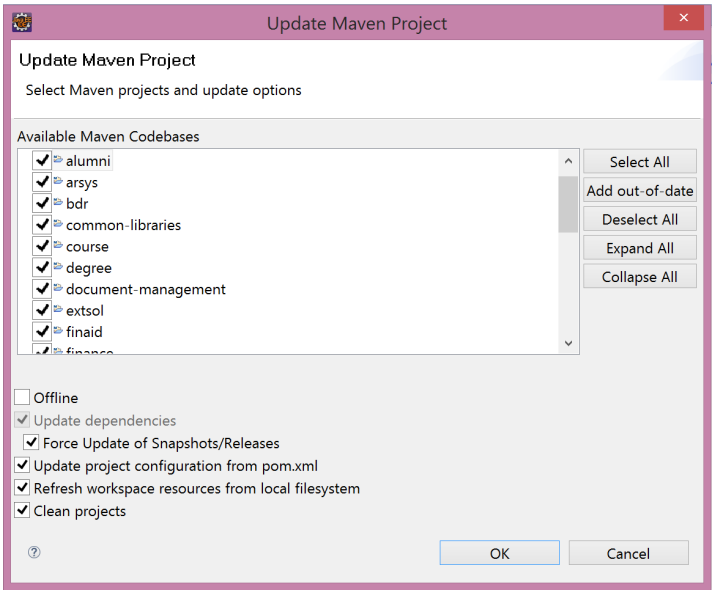
```
<dependency>
```

```
        <groupId>com.oracle</groupId>
        <artifactId>ojdbc7</artifactId>
        <version>12.1.0.1</version>
        <scope>provided</scope>
</dependency>
<dependency>
        <groupId>com.oracle</groupId>
        <artifactId>ucp</artifactId>
        <version>11.2.0.4</version>
        <scope>provided</scope>
</dependency>
```

Building your App

The build process compiles all pages and runs the frames framework to create the HTML pages.

1. Right-click on one of your projects and **select Maven > Update Project**.
2. Click **Select All** to choose all projects. (If you make changes later that require a Maven update, then on subsequent runs you only need to select the projects impacted.)

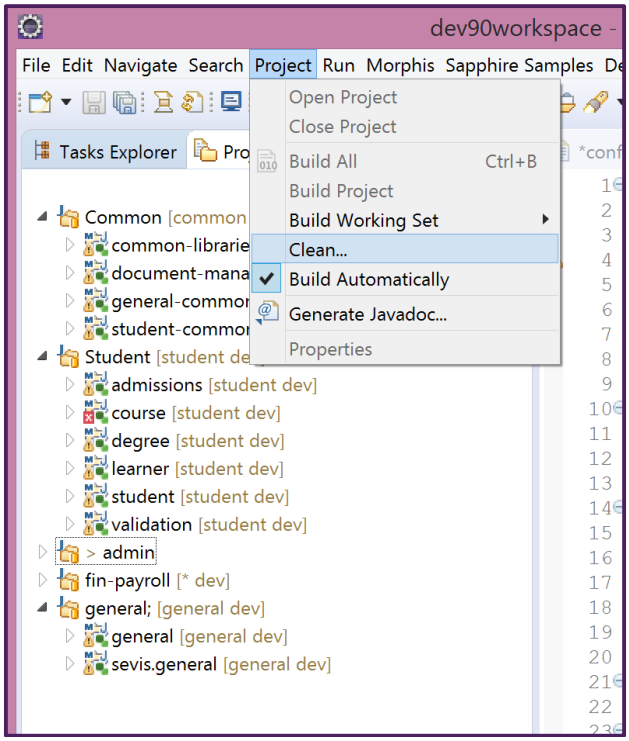


3. If more updates are required, save and then perform a **Project Clean**.

   You only need to perform a project clean if you have made many changes to the source. The clean process removes cached classes and frames objects.

   3.1. Access the Eclipse tool bar and selecting **Projects > Clean**.

   It may take a while to complete the maven update the first time.

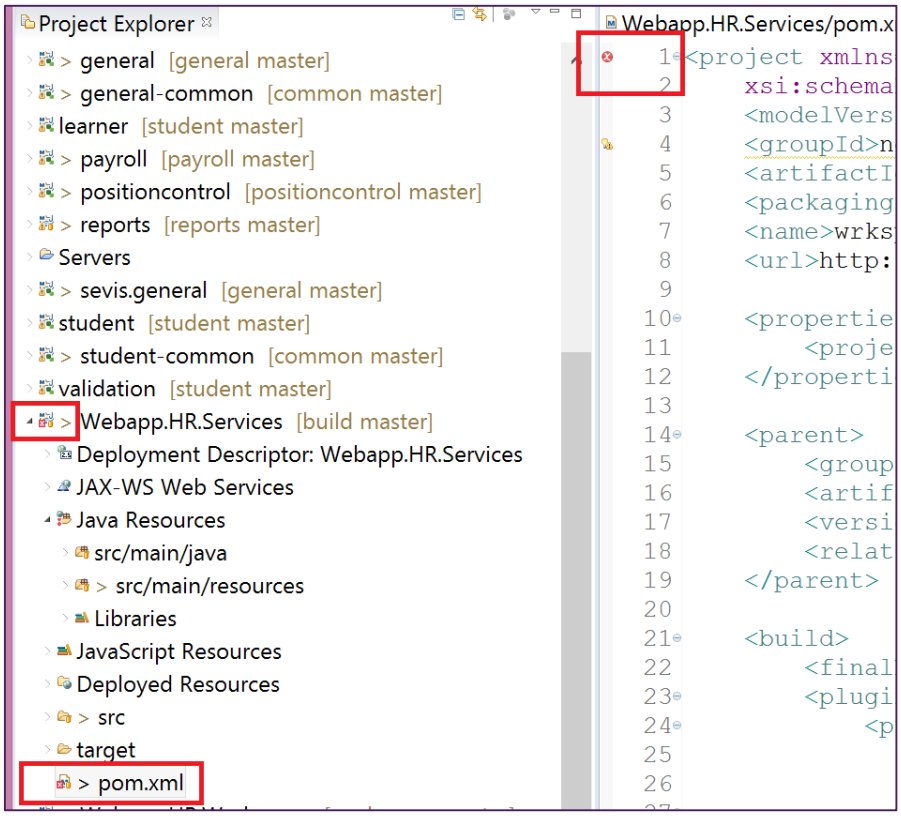


   3.2. Select to clean all projects.

Pom dependency errors

Missing dependencies or dependencies with missing imported projects are a common problem. The build will fail and the app won't run if there is an error.

You should always check for errors with the pom files. The red x shown in the following image indicates there is an error in the pom. The error needs to be corrected before the build can continue.



Optional Build Method

The following Maven commands, made from the command or terminal window, may be helpful when building the app.

```
cd c:/admin_page/dev90/workspace/webapp-workspace
mvn clean install

cd c:/admin_page/dev90/build/net.hedtech.banner.hr
mvn package
```

Running your App

Use the following steps to start the Tomcat server from the Servers view and run the application.

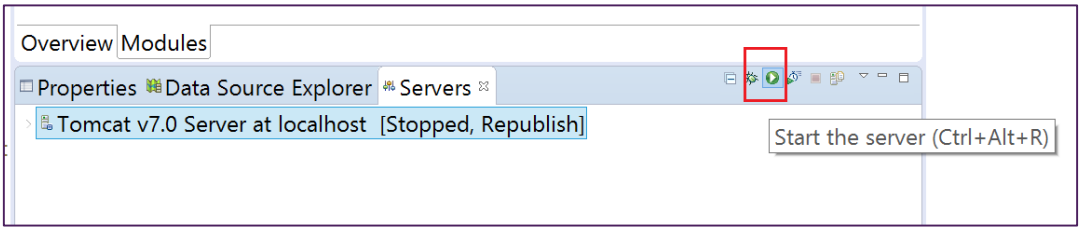
1. Navigate to the following URL:

   `http://localhost:8080/wrksp`

To start the app:

2. Click the Tomcat server and then the start icon.

   Note: Use the bug icon is to start the app in debug mode; use the arrow for a normal start.

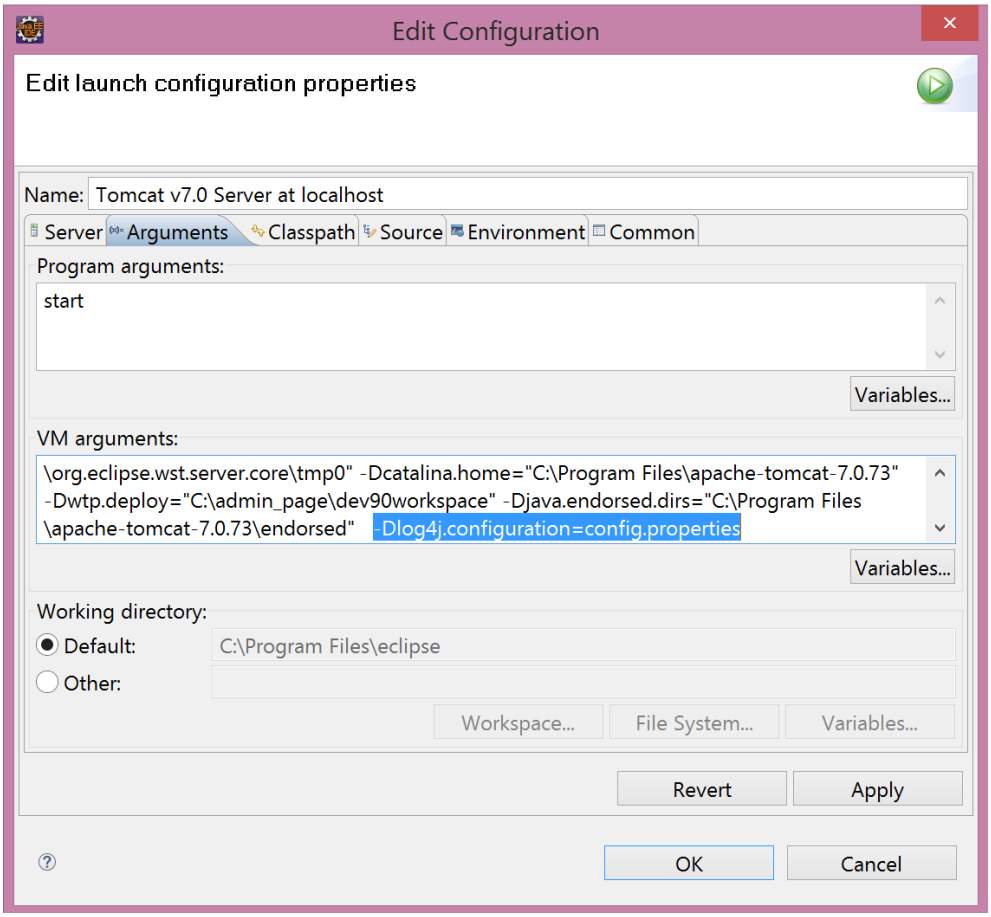


Logging

Use the following steps to enable logging for your application so that you can see all execution steps and error messages in the console.

1. From the Eclipse **Tool** menu, select **Run > Run Configurations** or click the **Open Launch Configuration** on the server overview properties.
2. Select the **Arguments** tab
3. Add the following arguments to the end of the **VM arguments** entry.

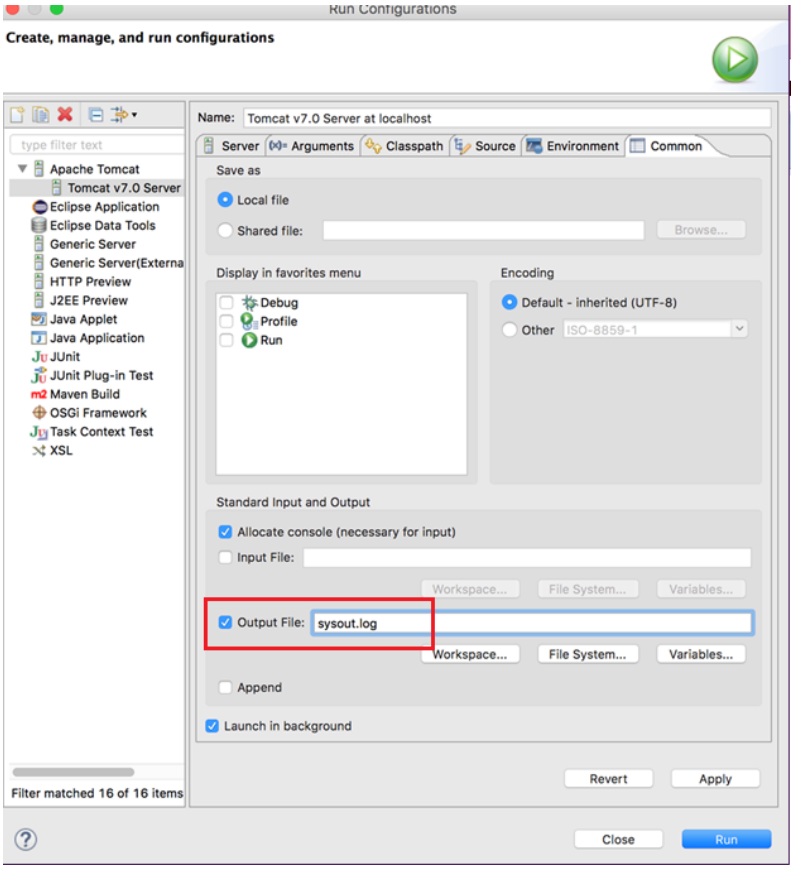
   ```
   -Dlog4j.configuration=config.properties
   ```



Do the following to specify a file to write the console messages to:

4. Select the **Common** tab.
5. Check the **Output File** checkbox and enter a file name.

   The output file will be written to the `c:\admin_page` directory.

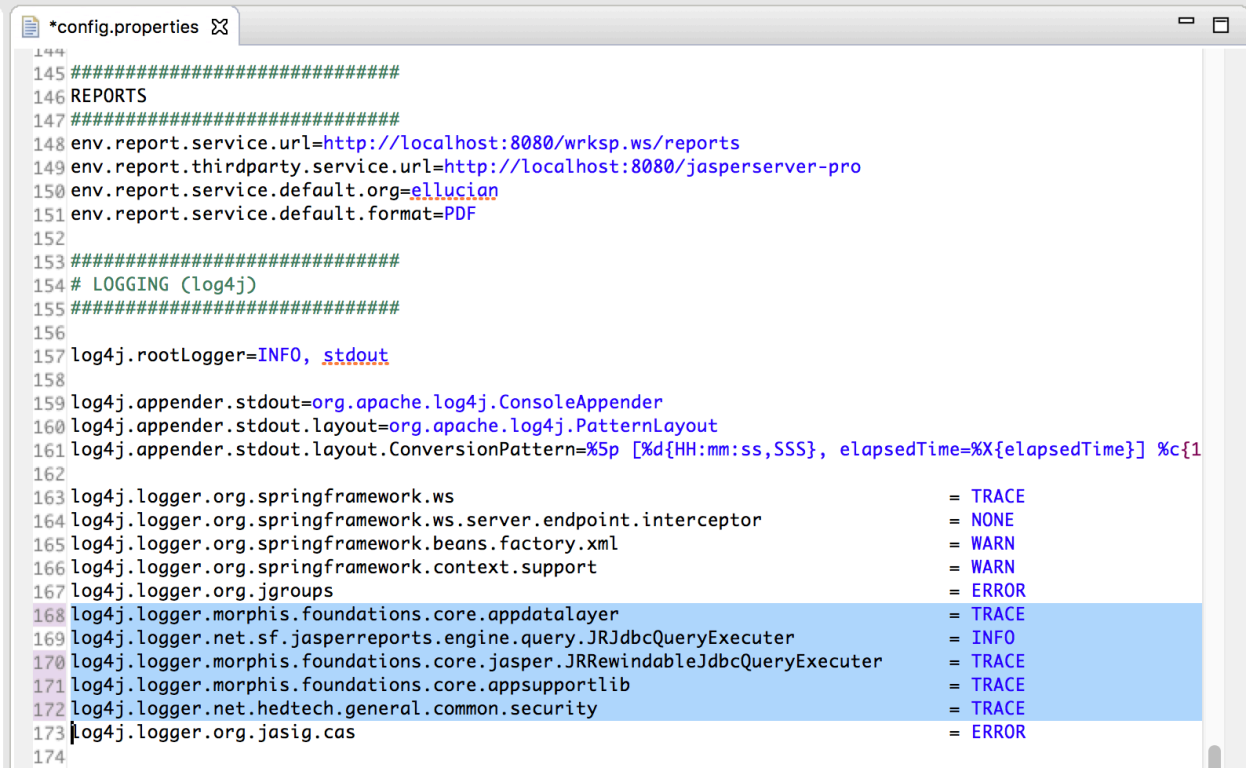


6. Set the log4j options to TRACE to maximize the feedback.

   The TRACE option means that debug, error and warn messages will show in the console.

   6.1. Open the `config.properties` file located the your **Webapp.HR.Services project** in the `src/main/resource` directory.

   6.2. Set the following highlighted packages to **Trace**.



Foundation Eclipse Tools

There are two tools that are useful in building pages: the GUI tool for building the UI and the page Task Explorer.

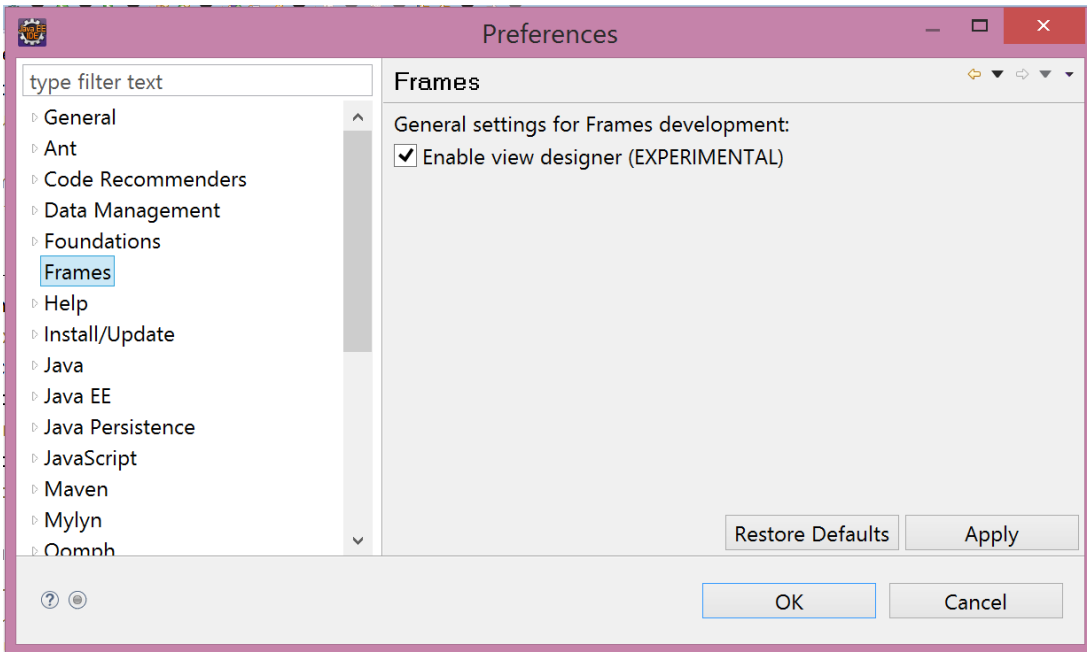
UI Development Tools

Use the following steps to enable the Frames Development.

1. Select **Windows > Preferences > Frames**.

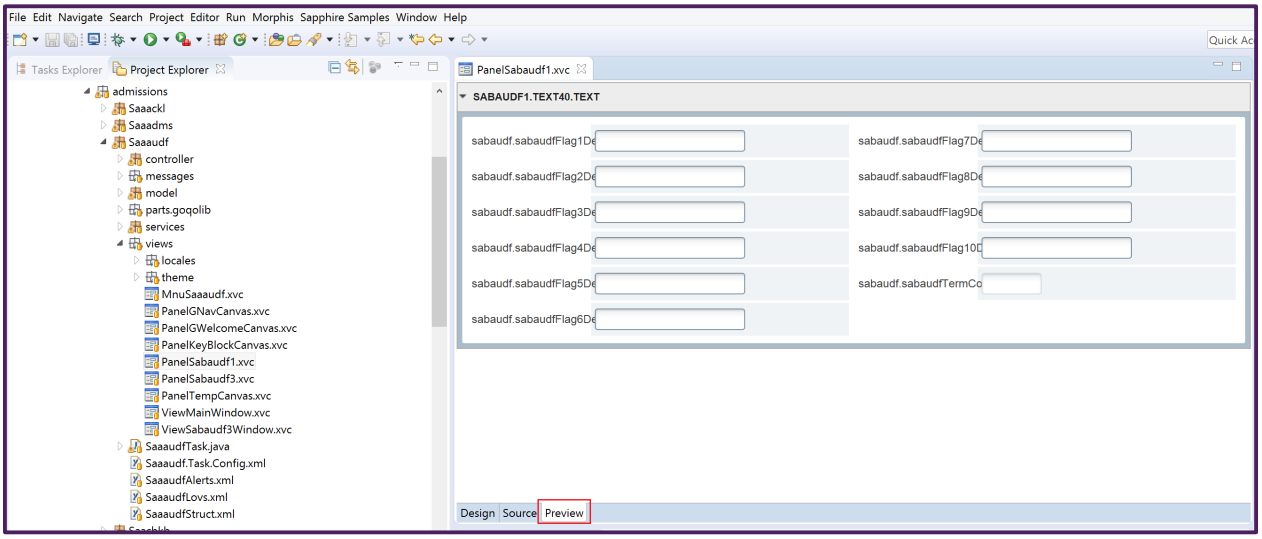
2. Under Frames, check **Enable view designer**.

   Note: The word experimental is displayed as a warning that the preview may not provide an exact view of how the page will look. The preview and design tabs are tools that assist with conceptualizing the page.

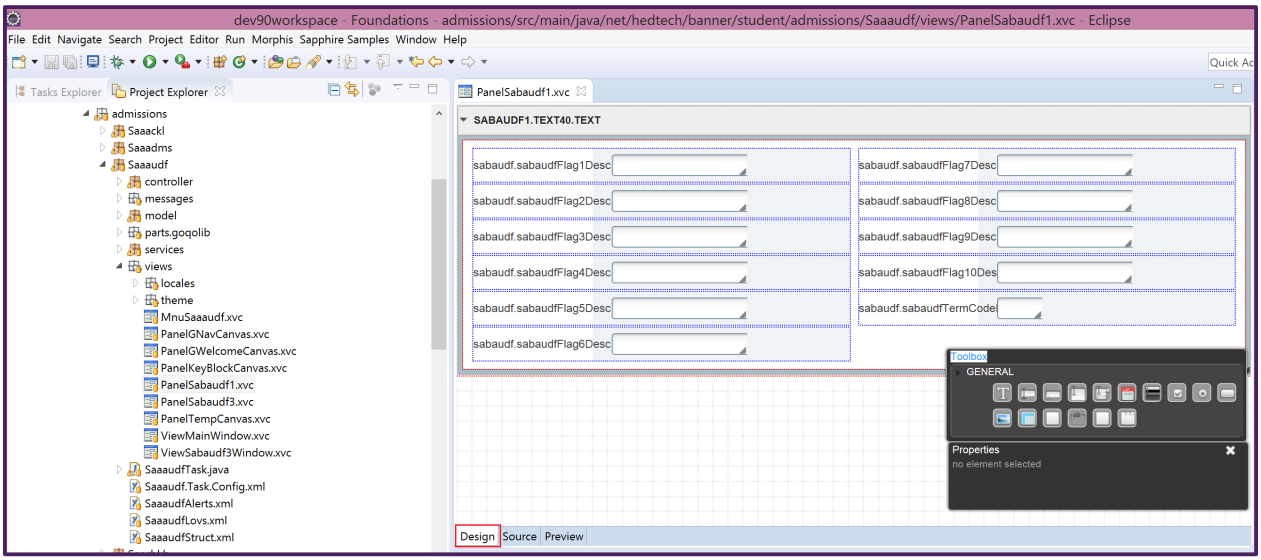


3. Open an .xvc file.

4. Select the **Preview** tab.



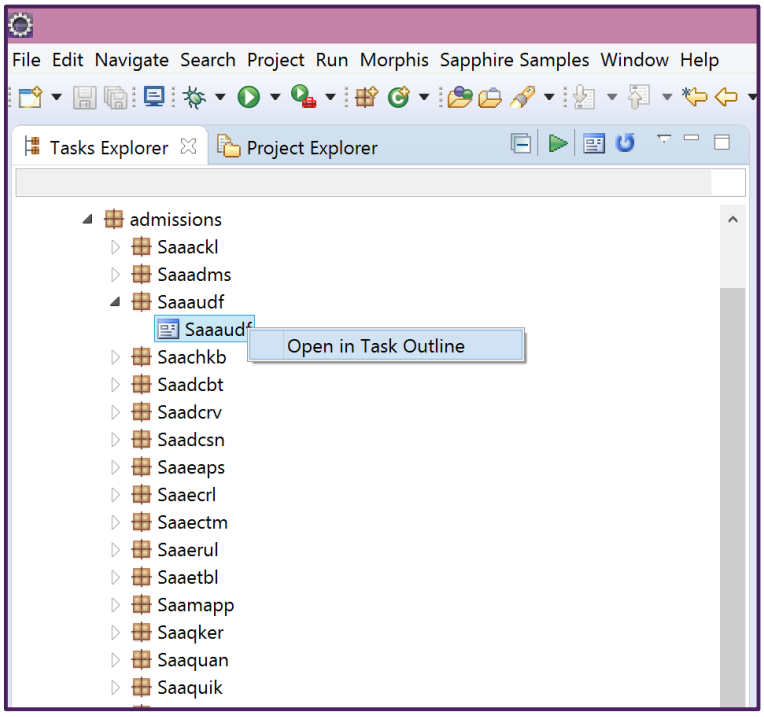
5. Select the **Design** tab to see the designer view.



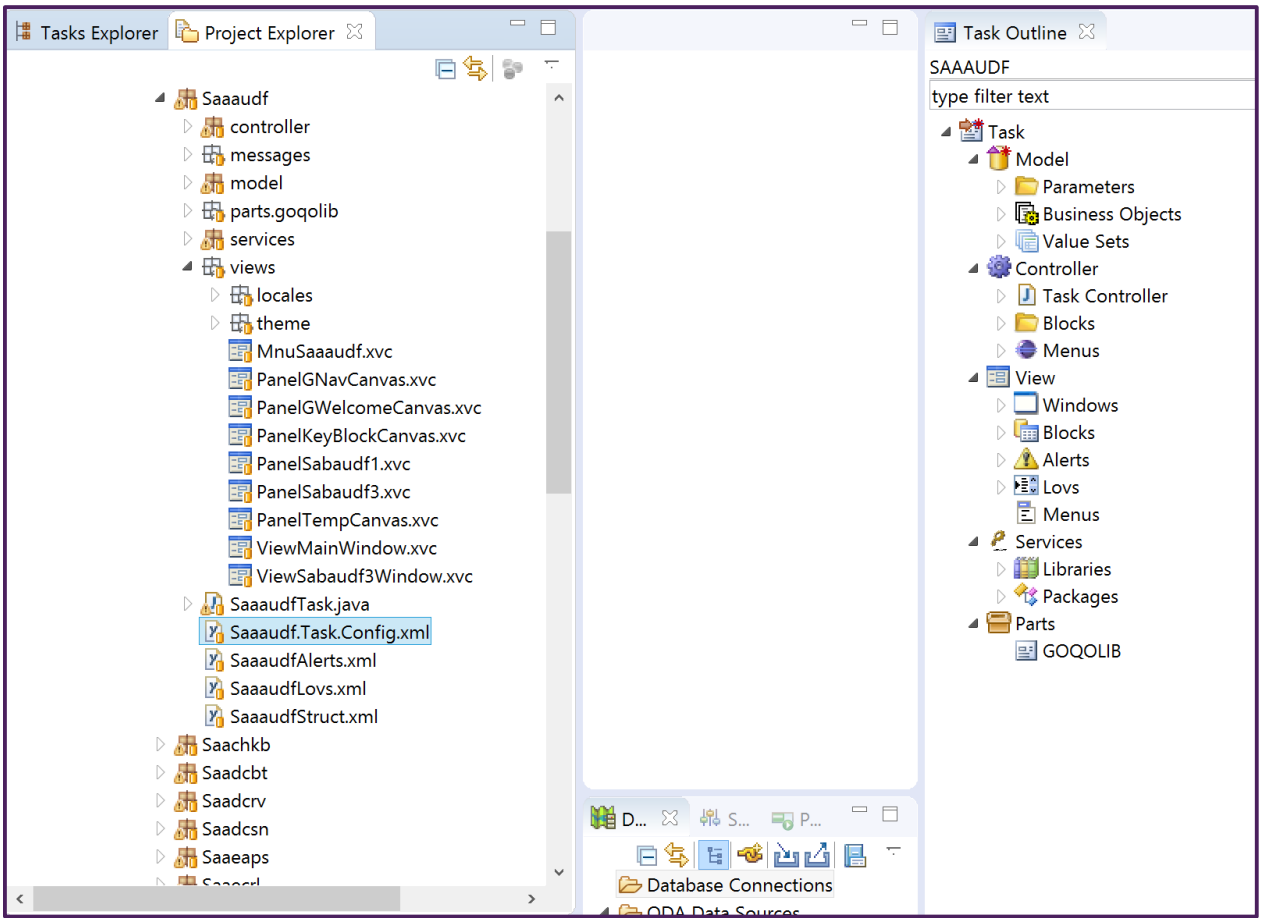
Task Development Tools

Foundations comes with an interface you can use for adding new objects to a page.

1. Expand the page contents in the Task Explorer.
2. Right-click on the page task and select Open in Task Outline.



The task explorer shows on the right side of the application.



See the *Banner Transformed: Extending Your Administrative Application* guide for more information about how to use the Task Outline with extensions.

Increasing the Apps Java Heap Size

Increase the Java Heap sizing in the following three places to improve performance: `JAVA_OPTS` environment variable, `eclipse.ini` file, and Eclipse Preferences. See the Java Heap Sizing recommendations section for guidelines about what size to use.

1. **`JAVA_OPTS` environment variable** – Increase the overall Java Heap size for your workstation located in the `JAVA_OPTS` environment variable.

2. **`eclipse.ini` file** – Increase the space allocated to Eclipse located in the `eclipse.ini` file.

    2.1. Find the `eclipse.ini` file.

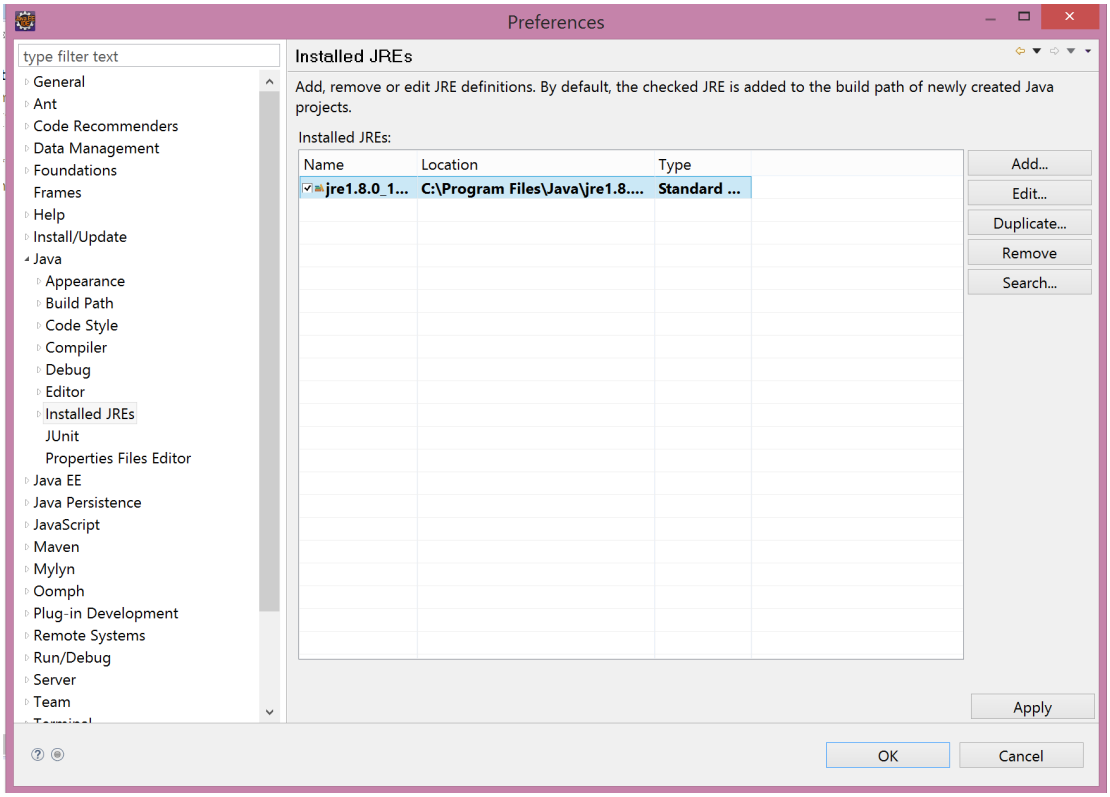
    - In OS, right-click on the Eclipse icon file, select show content, and find `eclipse.ini` in the expanded list.
    - On a Windows workstation, find `eclipse.ini` in the Eclipse folder.

2.2. Update the spacing and save the changes.

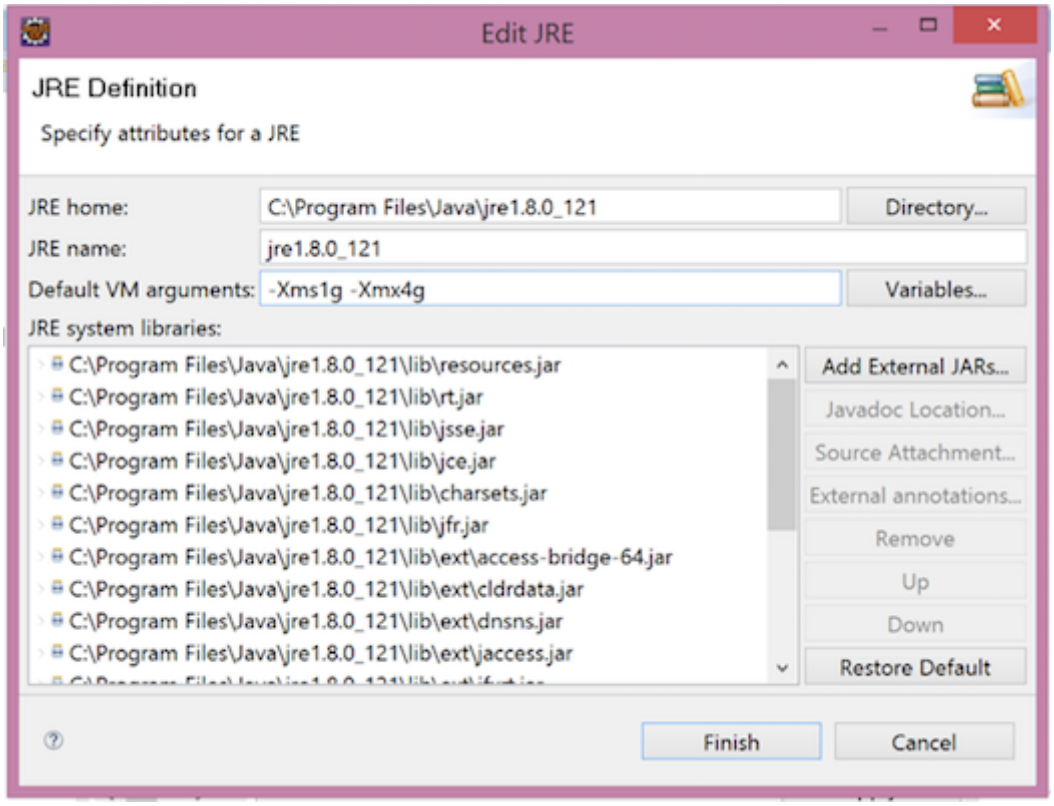
3. **Eclipse Preferences** – Update the space allocated to Java in Eclipse in the Eclipse Preferences.

3.1. Select Window > Preferences > Java > Installed JREs.

3.2. Select the JRE version you are running and click **Edit**.



3.3. Add the space arguments in the **Default VM arguments** field.



3.4. Click **Finish**.

Java Heap Sizing recommendations

The Java Heap size to use depends on the size of your workstation.

Use the following parameters for a workstation with large memory of 8-16 GB Ram.

-Xms1g -Xmx4g

If your workstation is smaller, try the following parameters:

-Xms512m –Xmx2g

The following table is a guide to the arguments.

Java Argument	Description	Example	Description
-XmsMM	Minimum amount of memory to use. Substitute MM for the size.	-Xms1g	Set minimum size to 1 gigabyte RAM. Use this value if your workstation 8-16 GB Ram.
		-Xms512m	Set minimum size to 512 megabytes RAM. 512 mg is lowest you will want to use and to be used if your workstation is less than 8 GB Ram.
-XmxMM	Maximum amount of memory to use. Substitute MM for the size.	-Xmx4g	Set maximum size to 4 gigabtyes of RAM. Use this amount if your workstation is 8-16 GB Ram.
		-Xmx2g	Set maximum size to 2G of RAM. Use this size if your workstation is less than 8 GB Ram.

Foundation Eclipse Tool Software Updates

Ellucian will deliver updates to the Foundation and Frames Eclipse plugins on a quarterly basis. The updates will include improvements to the Foundation Eclipse Development tools and to the objects used to compile pages. Do the following to receive updates:

1. Open Eclipse and select **Help > Check for Updates.**
2. If any updates are pending, Eclipse will prompt you to update.
3. Click **OK** to apply the updates.

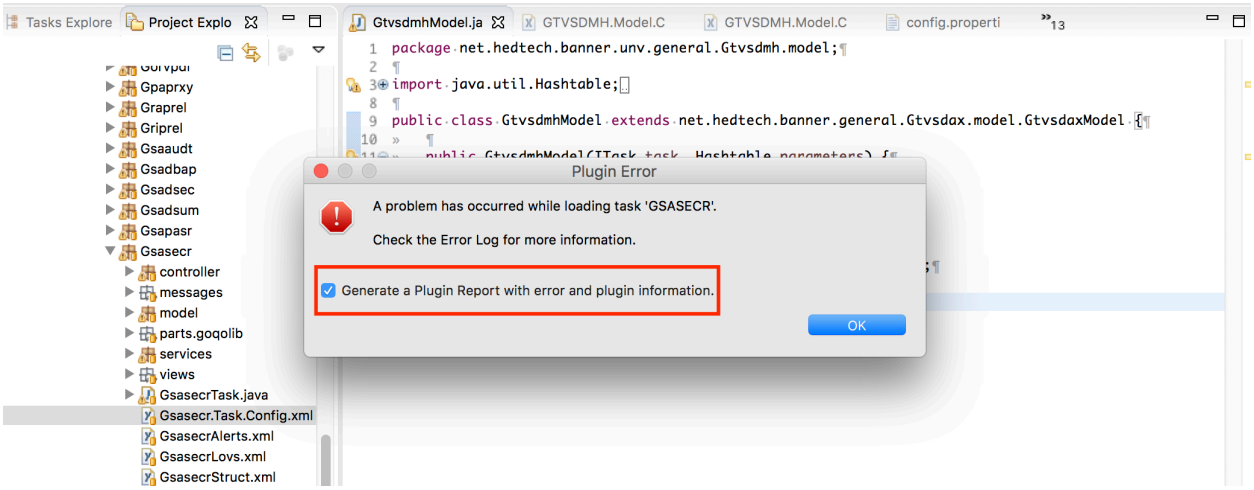
You may be required to restart Eclipse after the update is finished.

Generate Plugin Report for Morphis Plugin

If there are any errors that you encounter, then you can generate plugin report which can be used to see the errors in detail. These reports come in handy to detect and solve the errors.

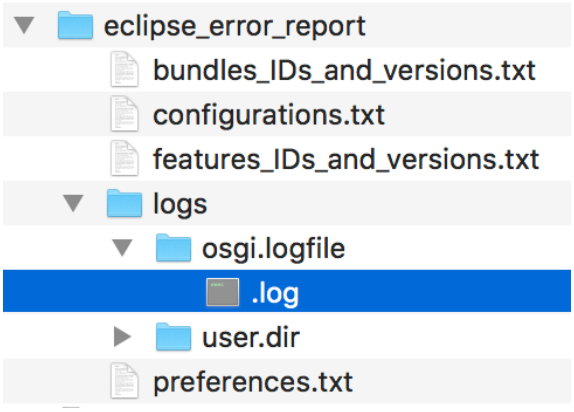
While you are modifying the application, there might be a scenario, when a plugin error will be prompted giving you an option to generate plugin report.

For e.g. in the below error prompt, you can check the checkbox to generate plugin report:



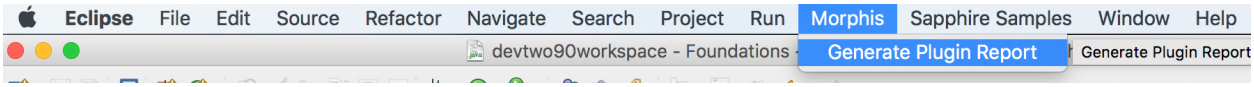
Next, in **Location** textbox you can give the path where you want the plugin report to be generated and then press **Ok**.

When you unzip the plugin report generated zip file, you will see the following structure:



Open the **.log** file which is located in **logs/osgi.logfile** folder. You will see all the generated errors in this file, bottom one being the most recent error.

However, there might be a scenario when an error does not prompt an option to generate a report. In that case, you can use **Morphis > Generate Plugin Report** option to create the report.



Apart from the log files, there are other files in the eclipse_error_report.zip which gives information about your eclipse environment, installed versions, preferences settings and other configurations which is also vital for error detection.