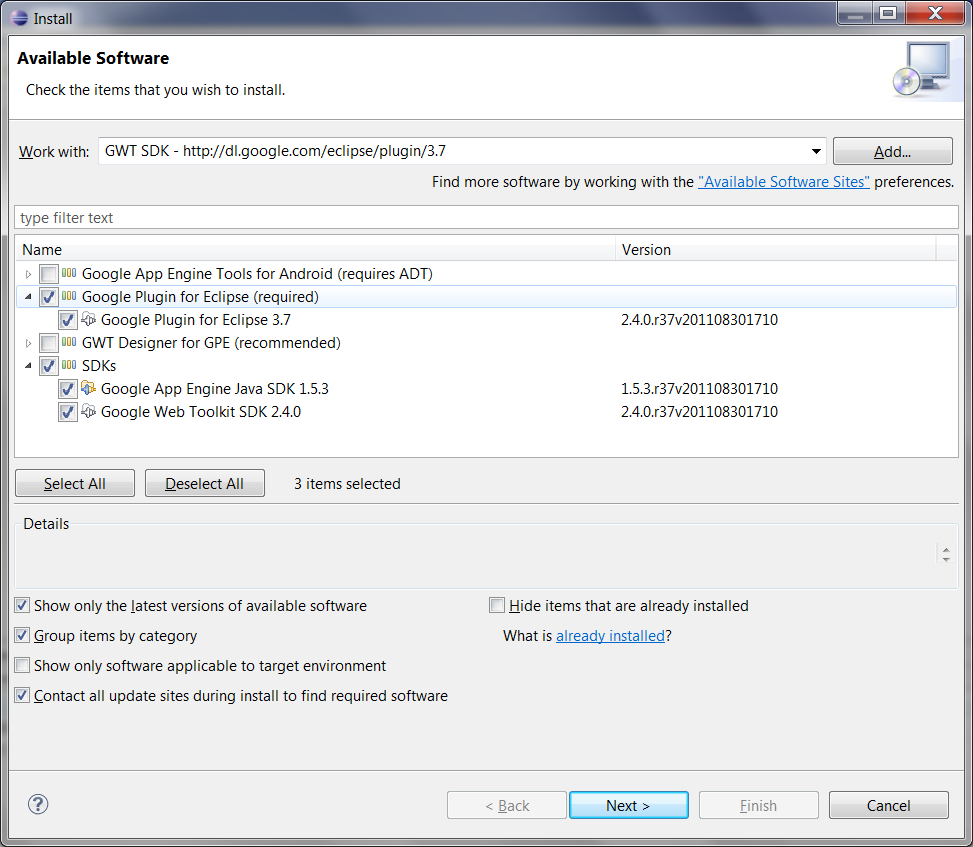
The WebStudio GWT project has been moved from experimental to the 5.1 branch under <http://svn.tibco.com/be/branches/5.1/webstudio/WebStudio>. This document describes what must be done to setup a development project to develop and run the GWT project from within Eclipse.

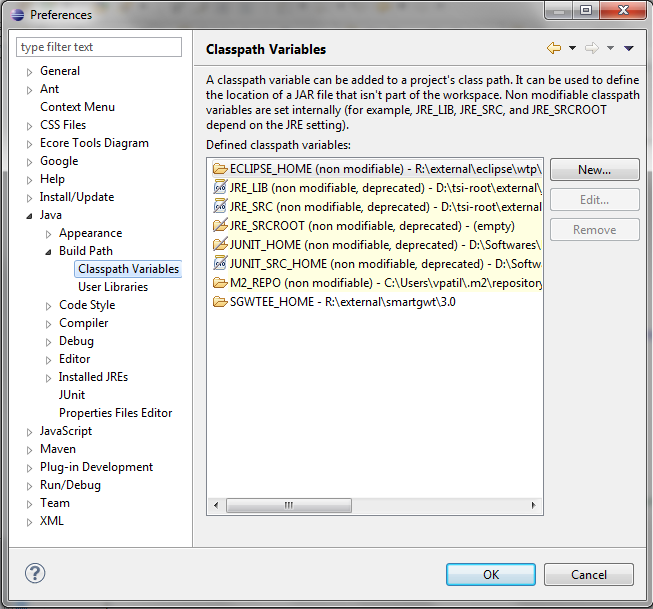
The first step that must be done is to install the Google Plugin for Eclipse. This step also installs the GWT SDK. The easiest way to do this is via the Google update site. For eclipse 3.7, the update site is <http://dl.google.com/eclipse/plugin/3.7>. Install the plugin and SDKs:



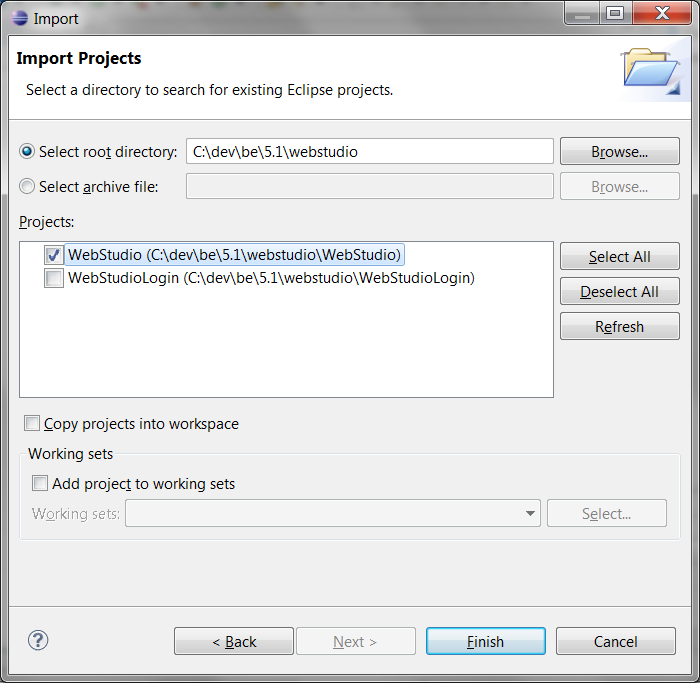
For more detailed information see <http://code.google.com/eclipse/docs/download.html>. As of this writing, the latest version of the GWT SDK is 2.4.0. After installing, it is best to restart Eclipse.

Next, you will need the SmartGWTEE library (library built on top of GWT) configured, if you have downloaded “tsi-root/external” on your local box, it should be located at “*tsi-root/external/smartgwt/3.0*”, else get it from [\\na-h-filer2\ql\tsi-root\external\smartgwt\3.0](file:///\\na-h-filer2\ql\tsi-root\external\smartgwt\3.0) and copy this archive to a local directory and unzip it.

Inside of your eclipse dev environment, you must now define a SGWTEE\_HOME classpath variable. Select Window->Preferences, and then Java->Build Path->Classpath Variables. From there, create a new classpath variable and point to the directory into which you extracted the SmartGWT archive.

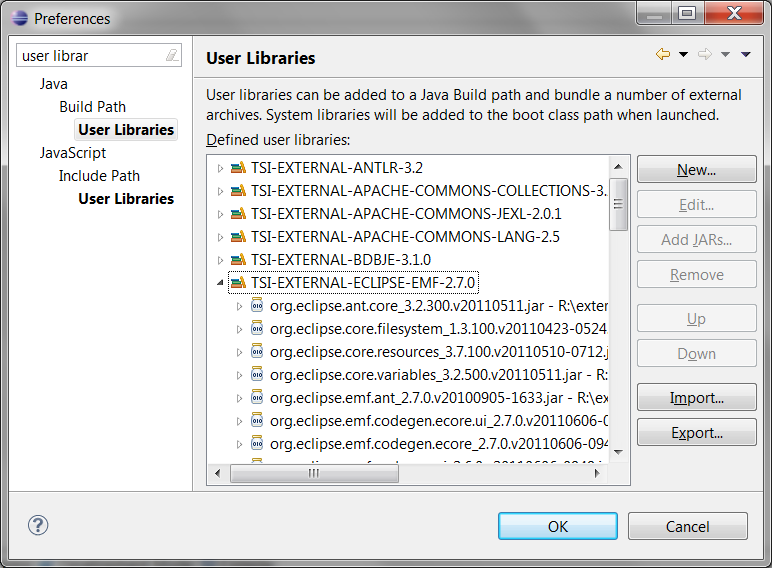


Now, update your 5.1 branch (or create one if you have not already done so). Import the existing WebStudio eclipse project found under “5.1/webstudio/WebStudio” by selecting File->Import->Existing Projects Into Workspace. Do not select “Copy projects into workspace”.



Make sure all the classpath references are correctly resolved. You will also need to copy gwt-fabridge.jar, gwt-servlet.jar, and gwt2swf-0.6.0.jar from [\\na-h-filer2\ql\shared\gwt](file:///\\na-h-filer2\ql\shared\gwt) into WebStudio/war/WEB-INF/lib.

The WebStudio project also requires EMF, which should already be installed in your eclipse environment (if not, the eclipse update site can be used to install the relevant features). If you have imported the eclipse.userlibraries file from 5.1/docs/setup-dev-env, you can add the TSI-EXTERNAL-ECLIPSE-EMF-3.7.0 user library to the project’s build path under the Libraries tab. If you have not yet imported this file, you can do so under Window->Preferences->Java->Build Path->User Libraries. Select ‘Import’ and select the 5.1/docs/setup-dev-env/eclipse.userlibraries file.

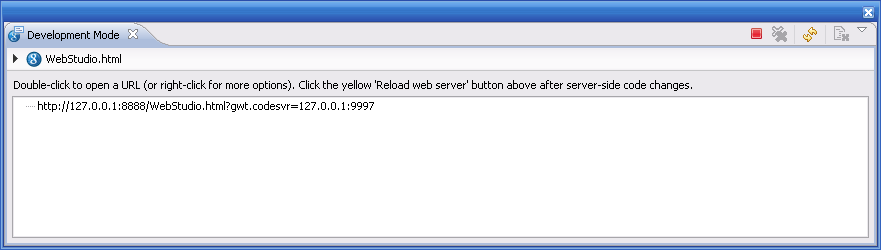


The WebStudio project should now build. If there are any build problems, consult the Problems view and resolve any build path/compilation issues.

For GWT compilation, either select GWT compile project from GWT Plugin or you can run ant target “build-webstudio”, this will copy all the jar and do a gwt-compile.

To run the WebStudio project in dev mode, simply right click on the project and select Run As->Web Application. The Google Eclipse SDK embeds a Jetty instance to serve up the servlets for debugging.

After the Web Application has started, you will see a URL in the Development Mode view:

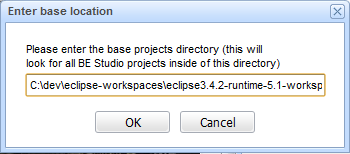


Select and copy the URL and open a browser (Chrome seems to work the best, from what we’ve seen, but other browsers should work as well). You might prefer to create a bookmark pointed to this URL if you work with WebStudio often.

Once the Web App loads, you will see a login screen. The user name/password is admin/admin.



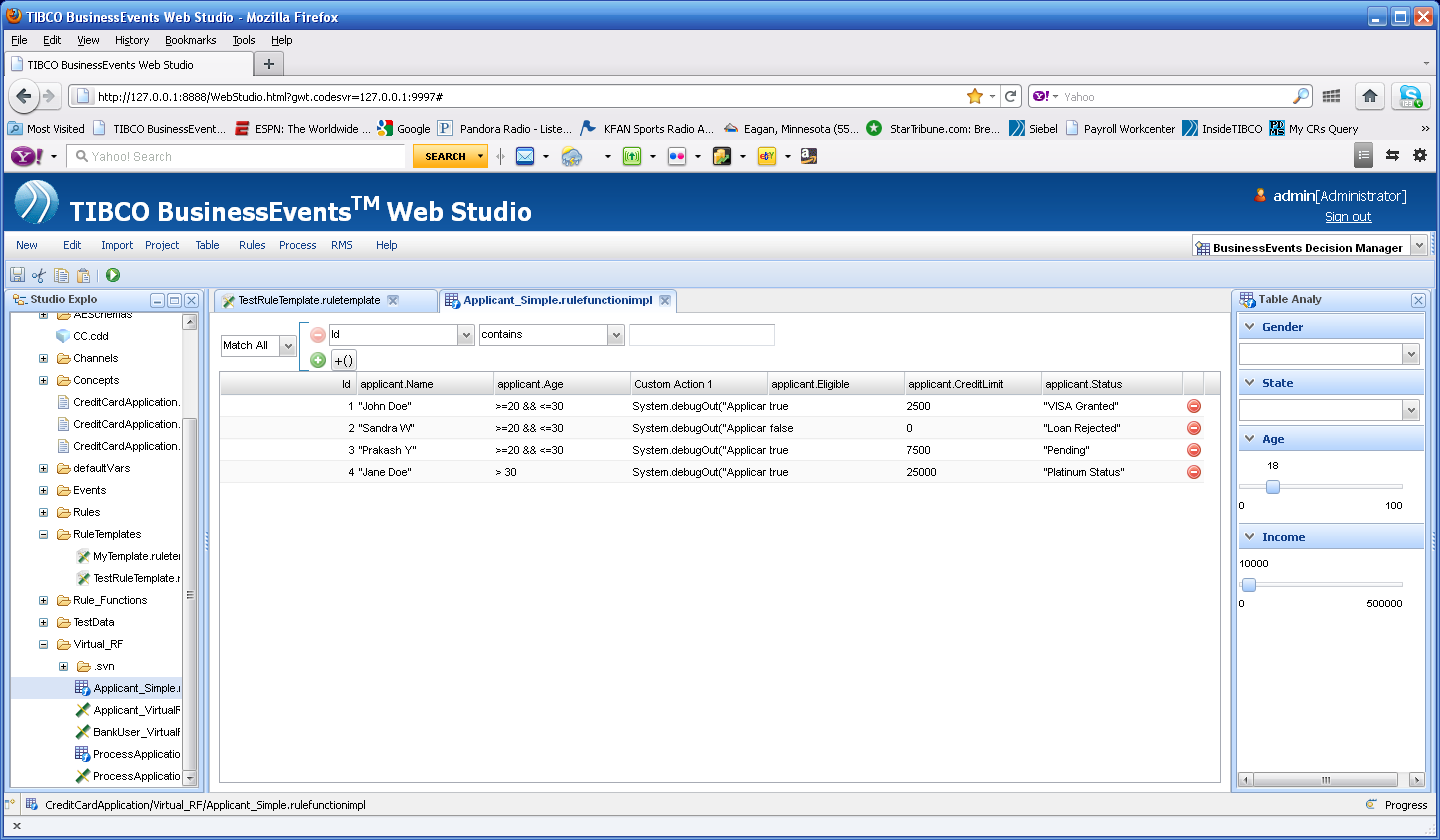
After login, you will be prompted to enter the base directory for your studio projects. This is a temporary step to allow us to continue development without “proper” client/server file management.



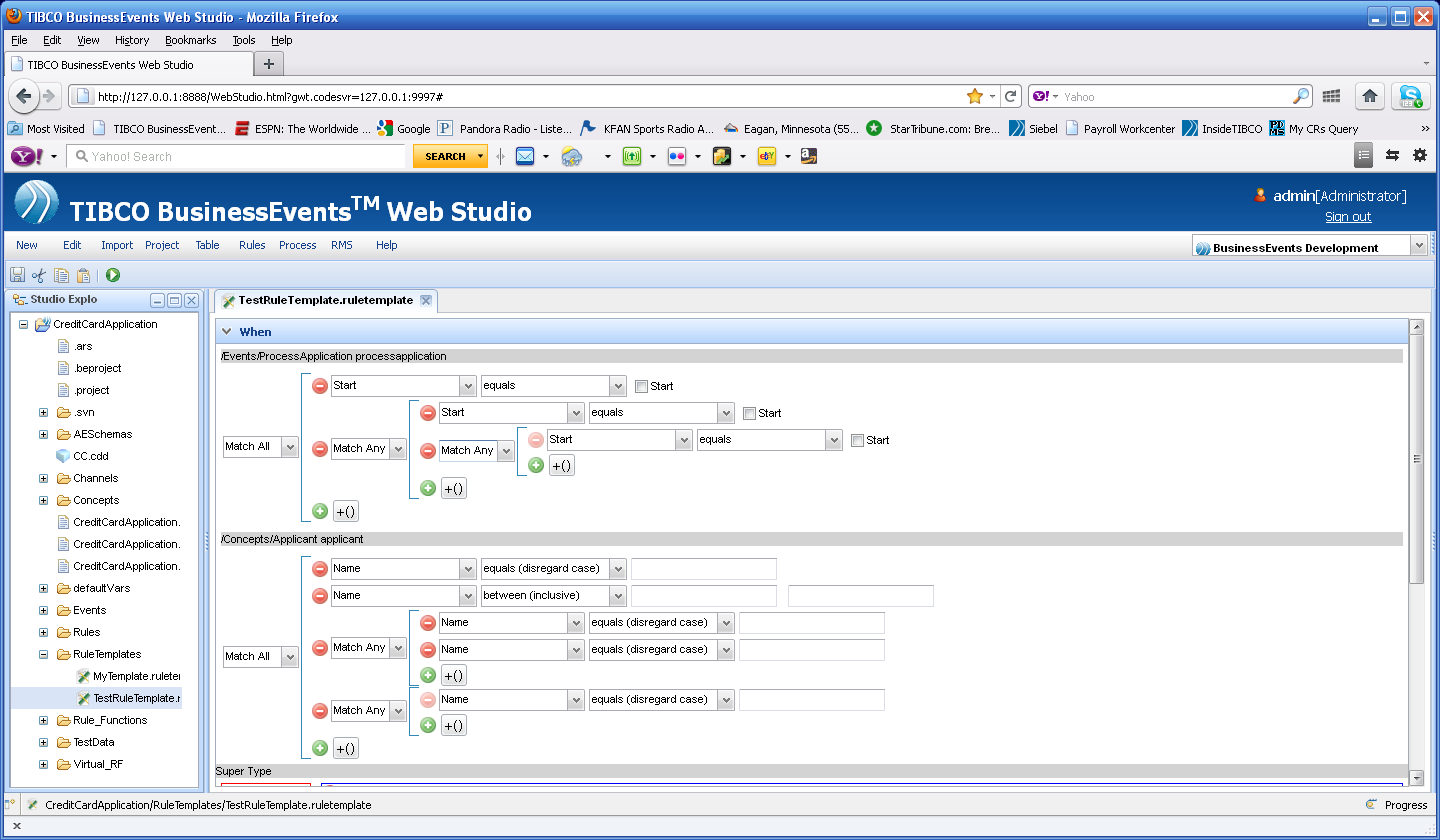
After that, you should see a “Home/Portal” like view.



If you click on an available project, it will take you to an IDE-like view. As of this writing, you can open decision tables



and rule template instances (which are created from rule templates)



More features will continue to be added along the way. To debug, simply start the Web Application with “Debug As” rather than “Run As”. Set breakpoints as you would normally, and you can debug the client/server side code.

If you have any issues, contact Ryan Hollom at [rhollom@tibco.com](mailto:rhollom@tibco.com).