**Micro RTS Installation Guide**

To obtain a copy of Micro RTS from Google, you will require a sub version client. In our research we used TortoiseSVN. TortoiseSVN is an Apache™ Subversion (SVN)® client, implemented as a Windows shell extension.. To acquire it for install on a Windows platform go to the link below

<http://tortoisesvn.net/downloads.html>

Once you have downloaded and installed TortoiseVSN, you can then download Micro RTS from google. Lets explain Subversion Systems. Subversion Systems also known as revision control system, and also known as version control systems permit the management of changes to documents, computer programs, large web sites, and other collections of information. The changes to the documents or projects are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", when changes are made revision 2 the resulting “set is revision 3” and so on. Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some and merged.

Today, the most capable (as well as complex) revision control systems are those used in software development, where a team of people may change the same files.

Version control systems (VCS) most commonly run as stand-alone applications. In computer science and software engineering, revision control is any kind of practice that tracks and provides control over changes to source code and the documentation associated with the source code. Professional Software developers use revision control software to maintain documentation and configuration files as well as source code.

As teams design, develop and deploy software, it is common for multiple versions of the same software to be deployed in different sites and for the software's developers to be working simultaneously on updates from various physical locations. For the purposes of locating and fixing bugs, it is vitally important to be able to retrieve and run different versions of the software to determine in which version(s) the problem occurs. It may also be necessary to develop two versions of the software concurrently (for instance, where one version has bugs fixed, but no new features (branch), while the other version is where new features are worked on (trunk).

In software development it has become increasingly common for a single document or snippet of code to be edited by a team. The members of the team may be geographically dispersed. Sophisticated revision control that tracks and accounts for ownership of changes to documents and code are extremely helpful and indispensable in such situations. This configuration refers to a subversion system named TortoiseSVN. TortoiseSVN is an Apache Foundation Subversion (SVN) client, implemented as a Windows shell extension for Microsoft Windows. It's easy to use, since it doesn't require the command line client to run and execute command. Additionally, you may want to familiarize yourself with Git and Git Hub. They are newer components available to one wanting to implement a subversion system. Git Hub is a Version Control System Repository Server that was chosen to house the components of this AI research project.

Micro RTS can be acquired at the following link

<https://code.google.com/p/microrts/>

Tortoise is windows shell extension. To use it you have to go through windows explorer. To access it you have to be on the desktop or in windows explorer. While at either of these two locations, right click your mouse. This will bring up a menu of selections. TortoiseSVN will be one of the items in the list. Point to the TortoiseSVN item in the list, this will open a list of TortoiseSVN commands. There will also be a selection for SVN checkout. You will use this option to checkout a version of Micro RTS from the Google Code Repository. On your system, create a folder that will receive the project transfer. After creating the folder that will hold the project, right click on the folder. Select VSN checkout from the list. This will display a dialogue window titles checkout. The dialogue window will have two textboxes to type into at the top of the window. One of the textboxes will have “URL of repository” as its label and the other will have “checkout directory” as its label. In the textbox labeled “URL of repository” type in the following URL text:

http://microrts.googlecode.com/svn/trunk/

TortoiseSVN may have already placed the text into the textbox for you if your browser has the Micro RTS site opened.

The textbox labeled “checkout directory” should contain the path to the folder that you have selected. In my case it read: “C:\Users\Grandmoff\Documents\SVNTEST” yours will probably be different

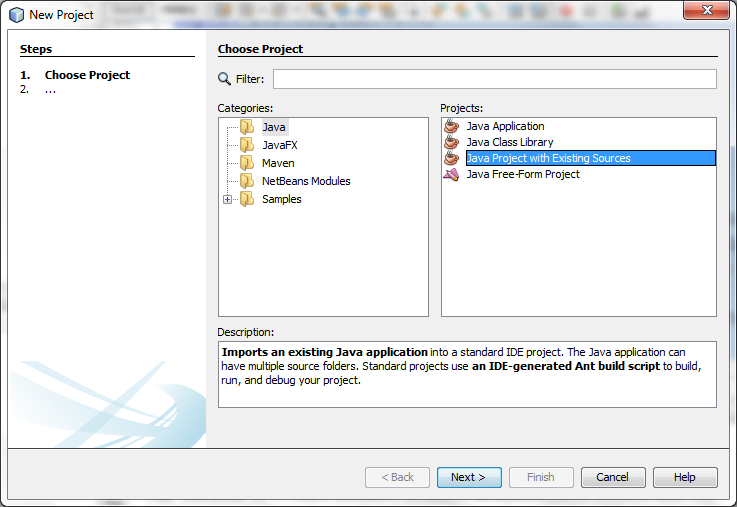
There will be a few other radio buttons and check boxes; you should leave these as they are for this project checkout process. When you press the OK button in the checkout dialogue box, the download will commence. When the download is complete, it will read completed at a certain revision number. Fine was 129, your may be different. Press OK to exit the dialogue box, the transfer will be complete. You can then move on to the process of creating a project and configuring your IDE to import the essential libraries to run the Micro RTS mouse driven component.

svn checkout **http**://microrts.googlecode.com/svn/trunk/ microrts-read-only

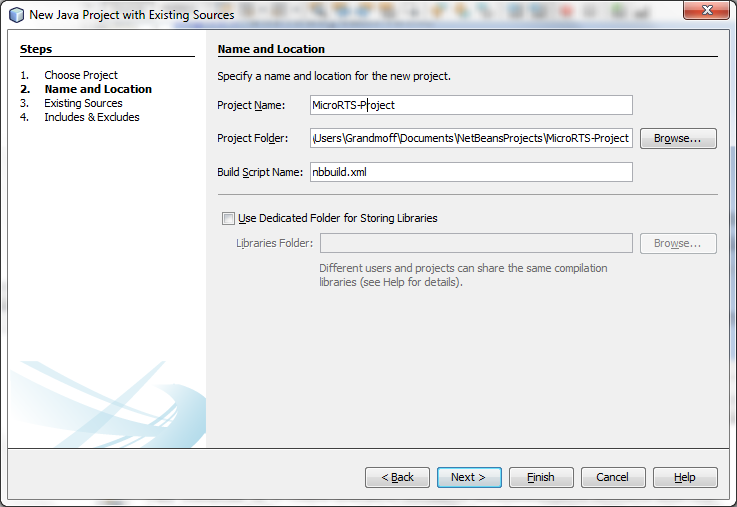
**In Netbeans, here is the overall abstract to create the project from the downloaded project folder**

1. Create a Netbeans project using the From existing project option
2. Add Micro RTS Source Folder to the newly created project
3. Add the Jdon.jar file using the add Jar file option in properties
4. Set proper working Directory to the imported project folder location

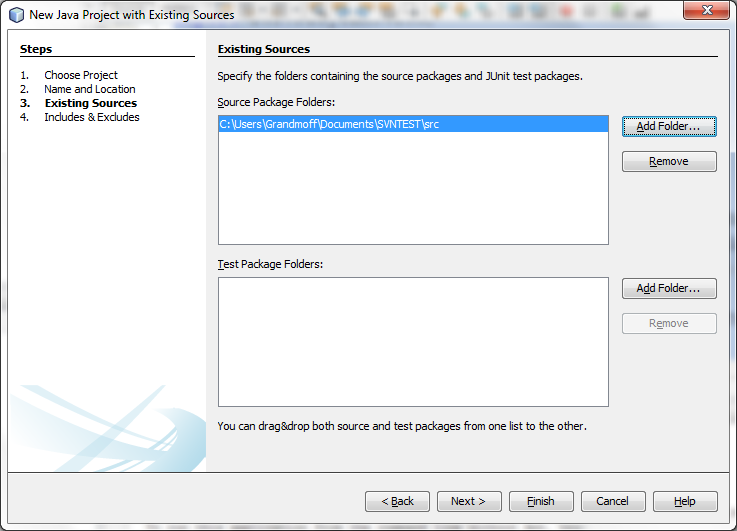
To Create a Netbeans project from an existing project, click on the file menu in Netbeans and pick new project. This will display a New Project dialogue box similar to the one illustrated below. Select the “Java Project with Existing Sources” as illustrated below. Click Next to continue



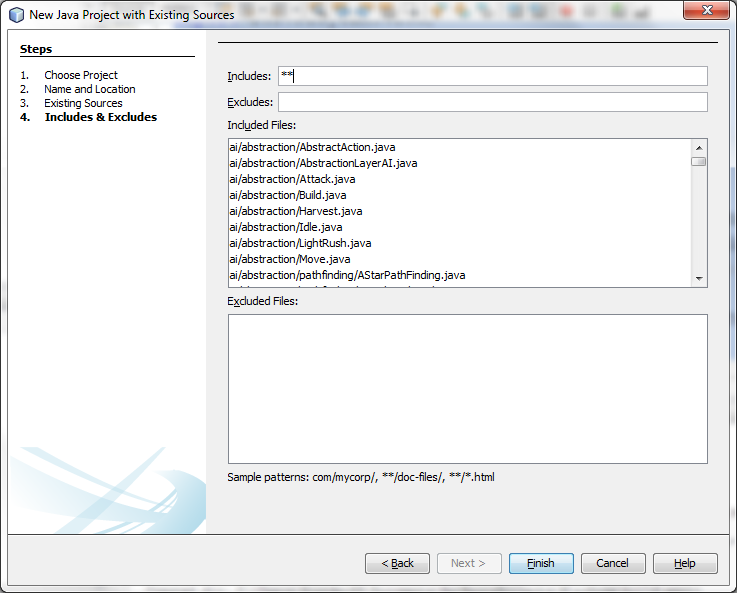
Select a Project Name and a location for your project folder in the Project Name and Project Folder text boxes. The Dialogue box is illustrated below. Click Next to go to the Dialogue Window



This will show you the “Existing Sources” Dialogue Box. In the source package folder area, select add folder. This will display a browse window. Use the browse window to navigate to the folder where you transferred the Micro RTS project to from the Google Code Repository. In my example I placed the downloaded project into a folder named SVNTEST. Netbeans will automatically pick the “src” source folder within the project.



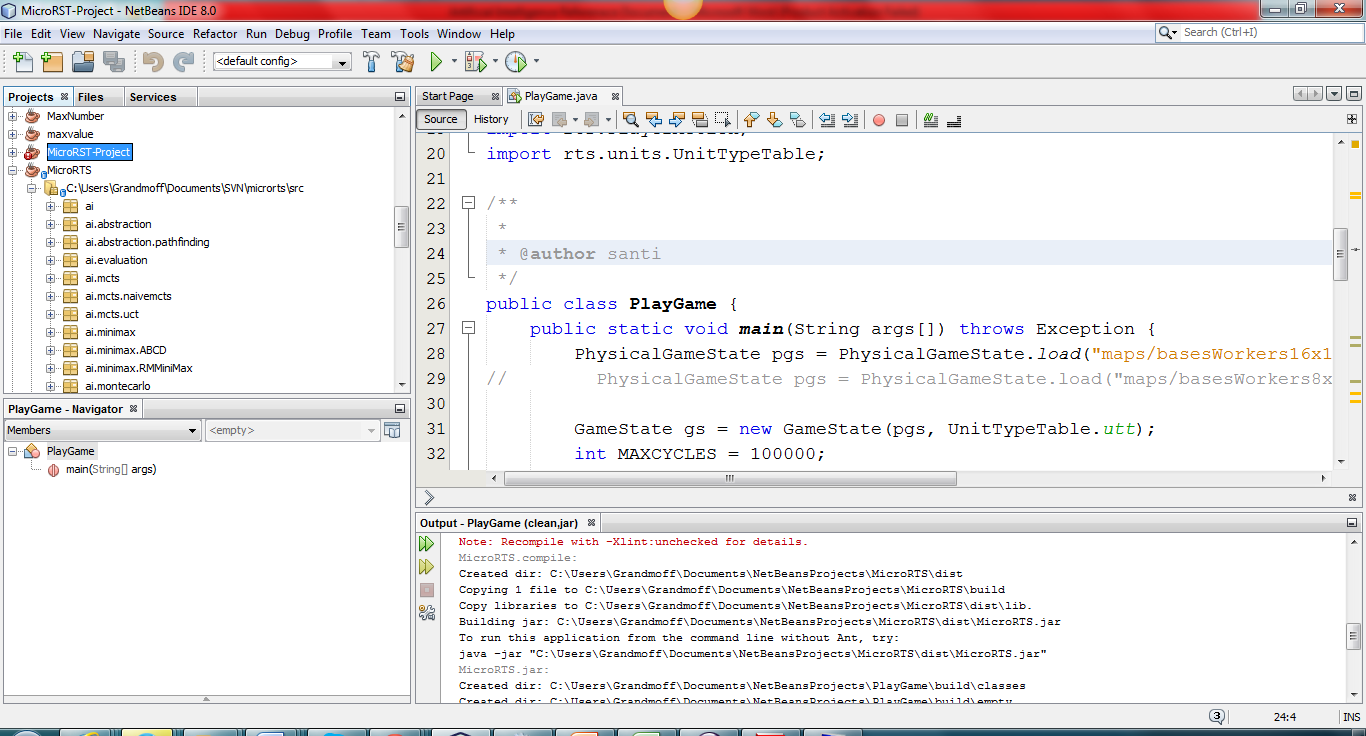
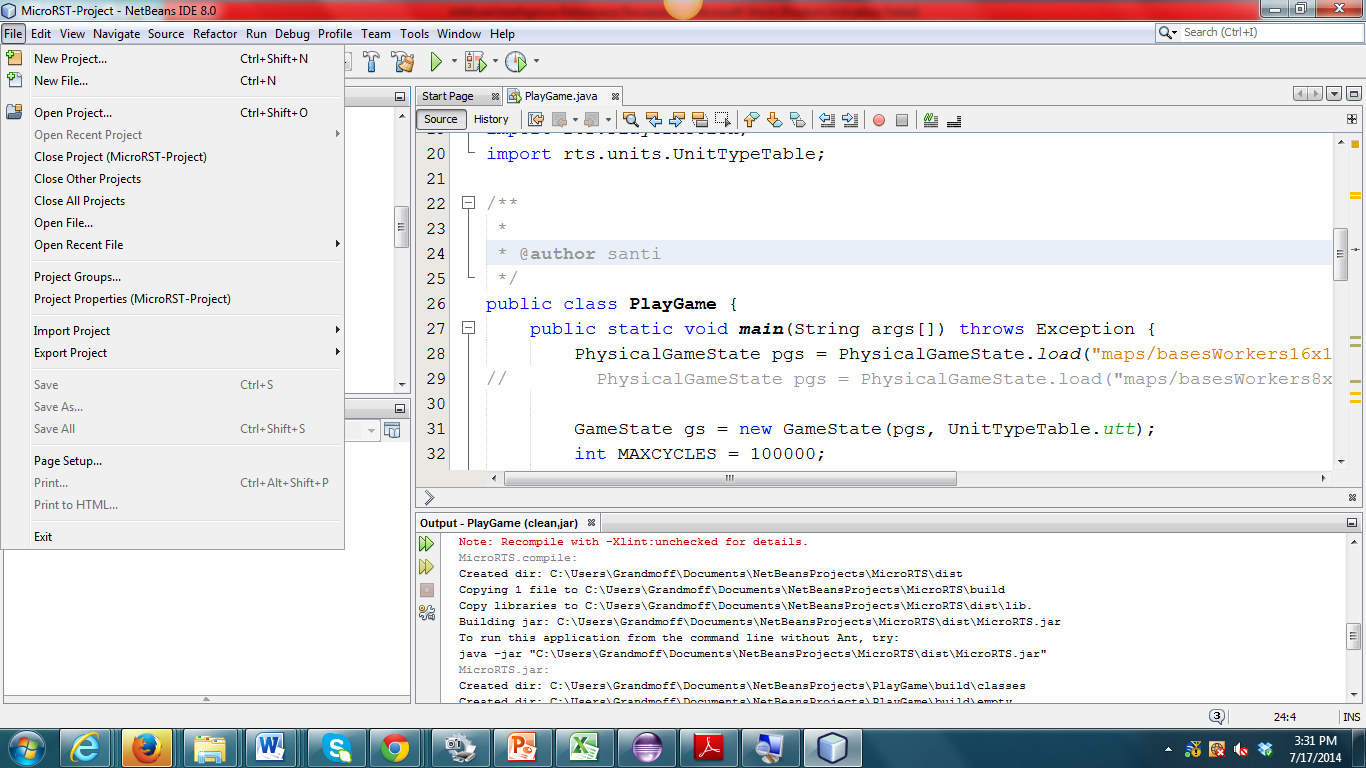
After clicking Next on the Dialogue box above a final include and exclude dialogue box will display, listing all of the classes that are going to be import ed from the Micro RTS project. Click finish



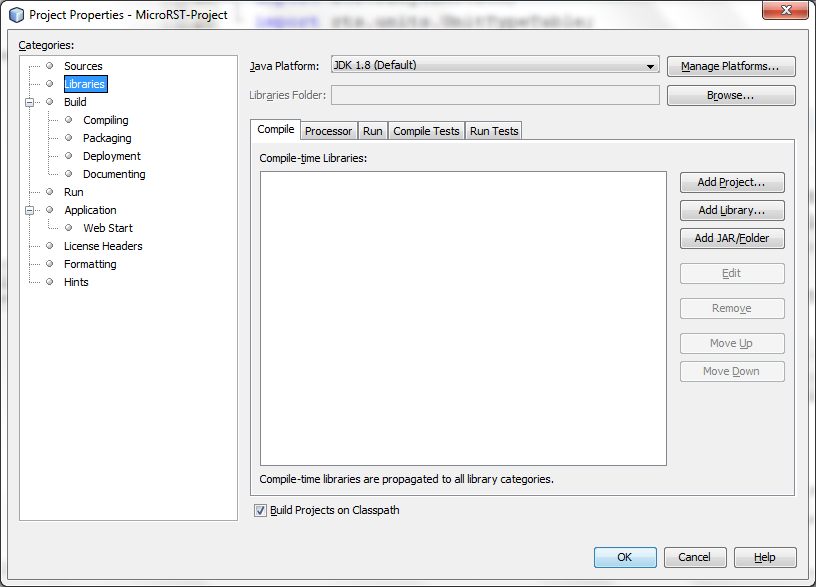
The source files will start to import; this will take several seconds depending on the speed of your system. After importing the code into your existing project, you can then begin to configure your Netbeans IDE for the game to run in user interaction mode.

A partially configured project folder is included with this lesson. It includes the playgamewithmousetest class as the main class in the project. This will limit the work that you will have to do to setup the playing environment.

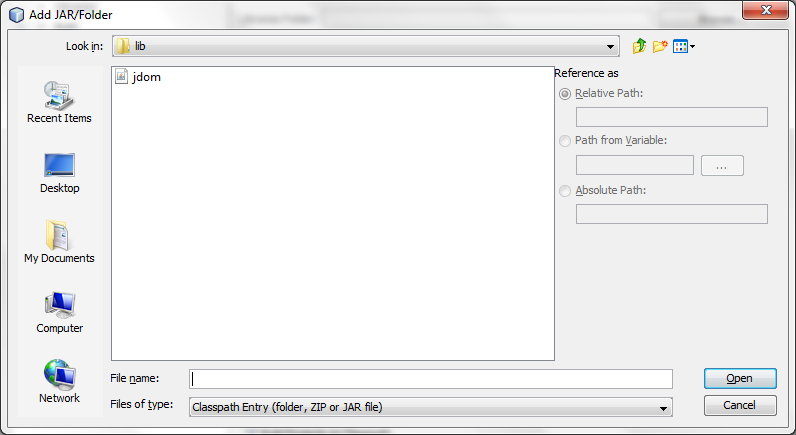
Now all you have to due is step 3 and 4, importing the jdon.jar library and setting the working directory to the location of the Micro RTS folder. This is the folder that you downloaded Micro RTS into. To import jdon.jar, click on the project in Netbean’s project navigator. The first screen shot below illustrates my project name “MicoRTS-Project” selected in the project navigation panel. The second screen shot shows the file menu with the “project properties” selection midway down the list. You should select the project properties selection.



In the project properties dialogue window, select libraries. This will display a window similar to the one illustrated below. Click on the add jar button on the right side of the screen

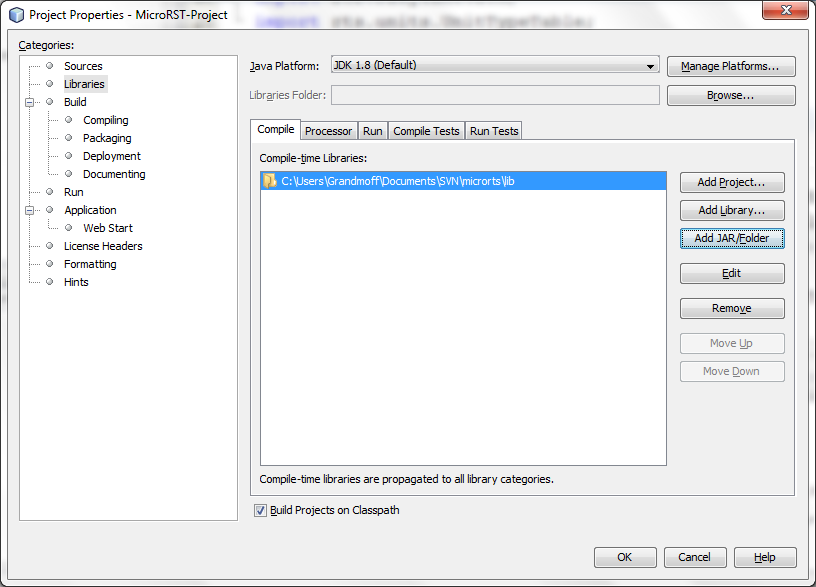


This display the add jar dialogue box as illustrated below.



Navigate to the jdon.jar file located in the Micro RST project library folder and click open

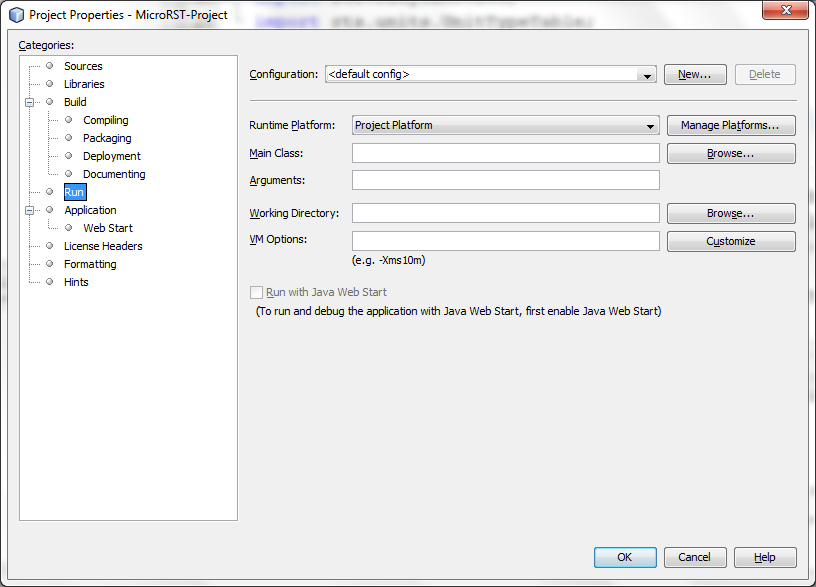
This will add the jdon.jar file to the collection of jar files that will support the application. It should be the only jar file in the list, as illustrated below.

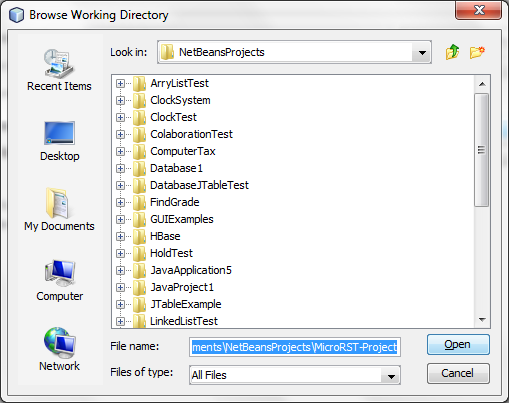


Click OK to complete the task

Finally you have to select your Micro RTS project folder as the default working folder. This is the folder that you named when you transferred a copy of the project from the Code Repository in Google using TortoiseVSN.

First you have to go back to project properties by using the pull down file menu and selecting project properties from the list. Just name as you did when adding the jdon.jar file to the project. This time instead of selecting libraries select run from the categories list on the left side of the window. This will display a dialogue window similar to the one listed below. Click on the browse button that is aligned with the label “Working Directory”





Navigate to select your Micro RTS project folder as the default working folder. This is the folder that you named when you transferred a copy of the project from the Code Repository in Google using TortoiseVSN. This completes the configuration, you should now be able to run Micro RTS