Using Eclipse

Using Eclipse

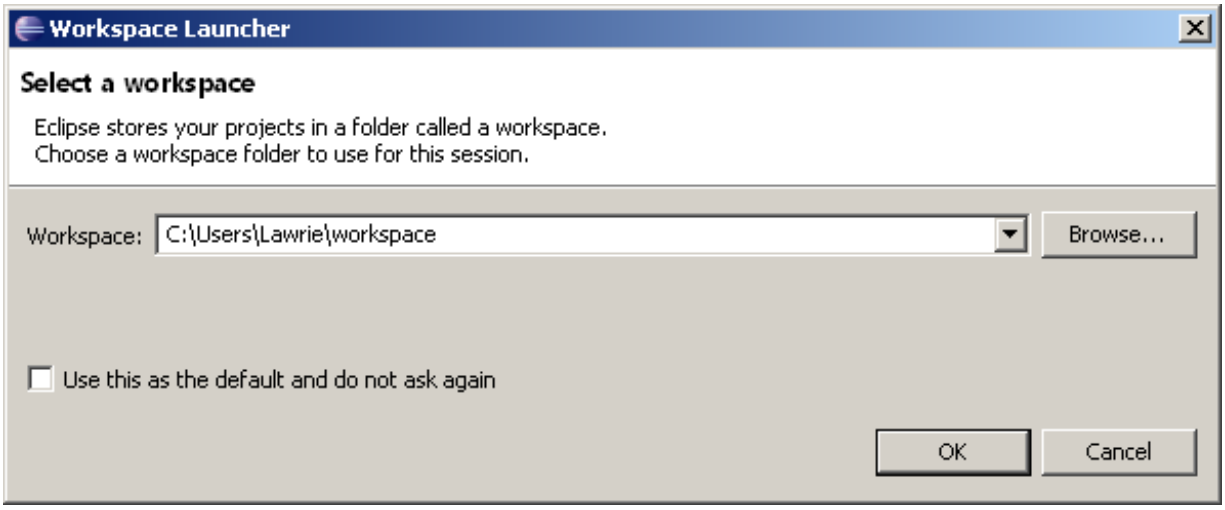
Installing Eclipse

You can download Eclipse from Eclipse downloads. If you are only going to use Eclipse for leJOS NXJ programs, you will only need standard Java development capabilities - you will not need a version that supports the Java Enterprise Edition or other languages such as C++ or Eclipse plugin development. A package such as "Eclipse IDE for Java Developers" is sufficient for leJOS NXJ. Make sure you download a package for your operating system. Note, however that for Windows, even if you are on a 64-bit system, you need a 32-bit version of Eclipse, for leJOS NXJ. The examples in this section are for Microsoft Windows, but the installation process is similar for other operating systems. If you are using Linux, your distribution may have its own Eclipse package, which you may prefer to use.

Unzip the package you downloaded to a folder. If you extract all files to C:\ then your Eclipse folder will be C:\eclipse.

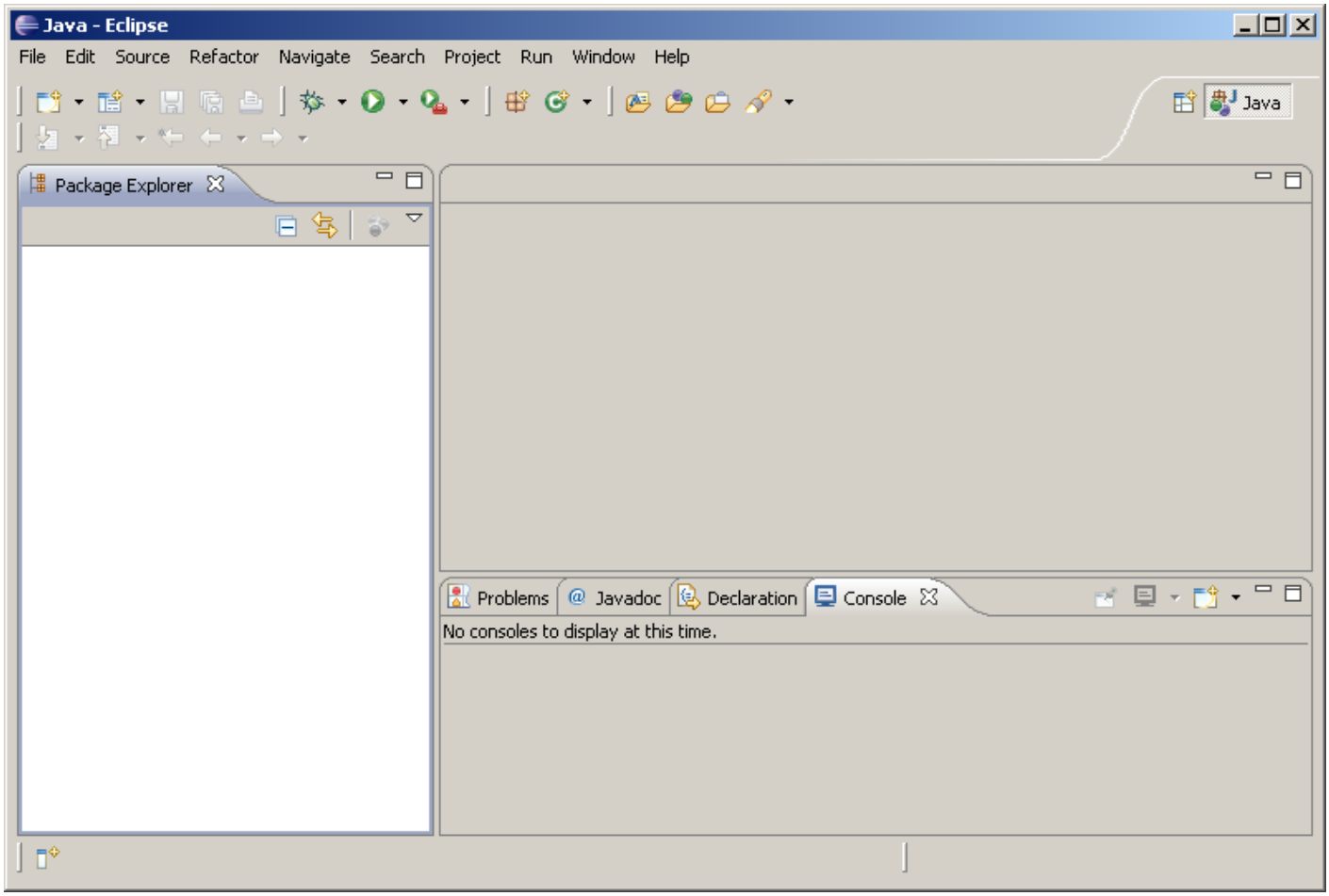
You can start Eclipse by running eclipse.exe from your Eclipse folder. You may want to put a shortcut to eclipse.exe on your desktop.

When you first start Eclipse, you are asked to select a workspace:



You can leave this as the default, or you can work with multiple workspaces and create one just for your leJOS projects.

When you click OK, the IDE will open and you will see a welcome screen. After closing the Welcome window, Eclipse will look like this:



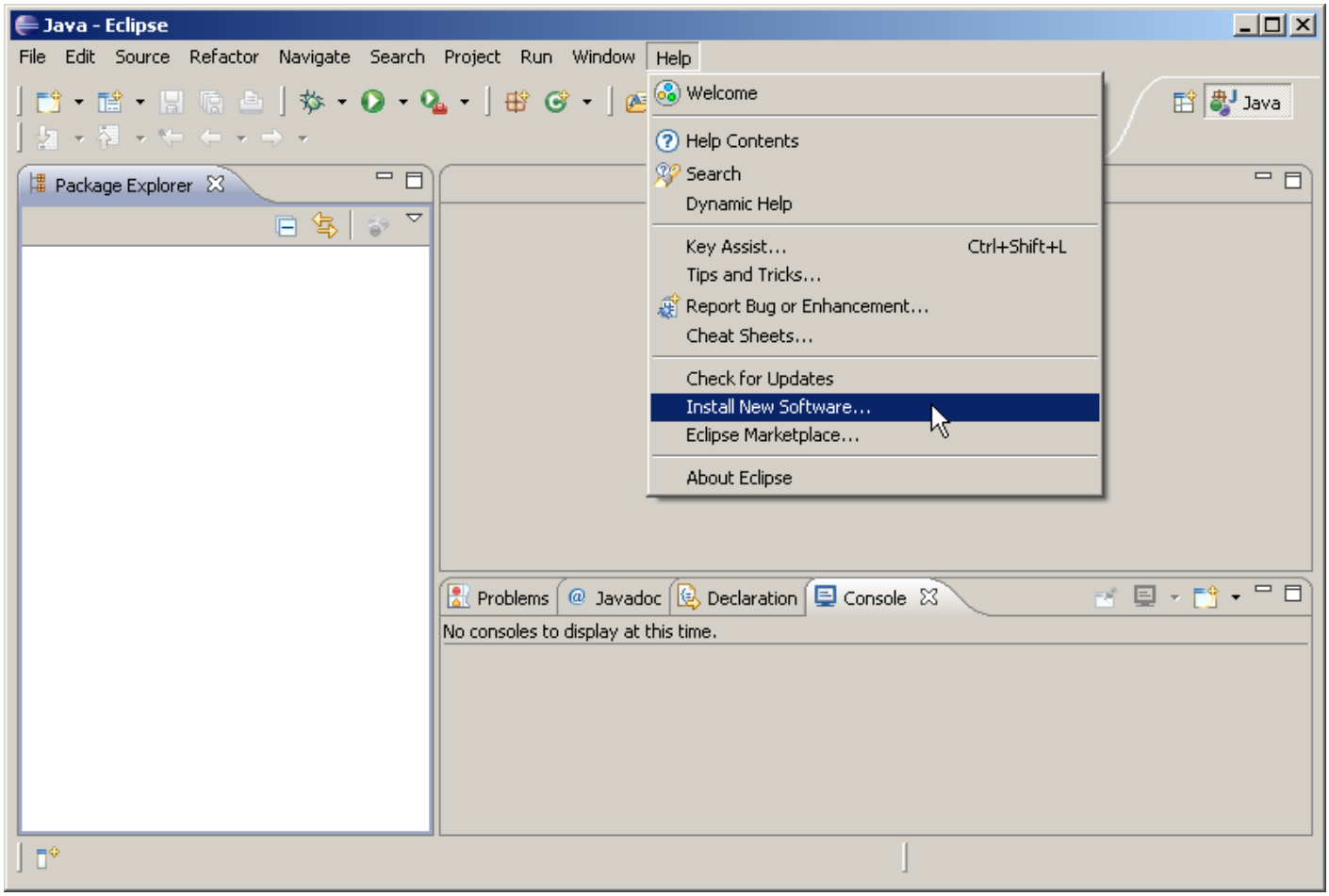
The tutorial takes you through installing the leJOS plug-in, creating your own NXT and PC leJOS project, and running the NXT samples and PC samples that come with leJOS.

Back to top

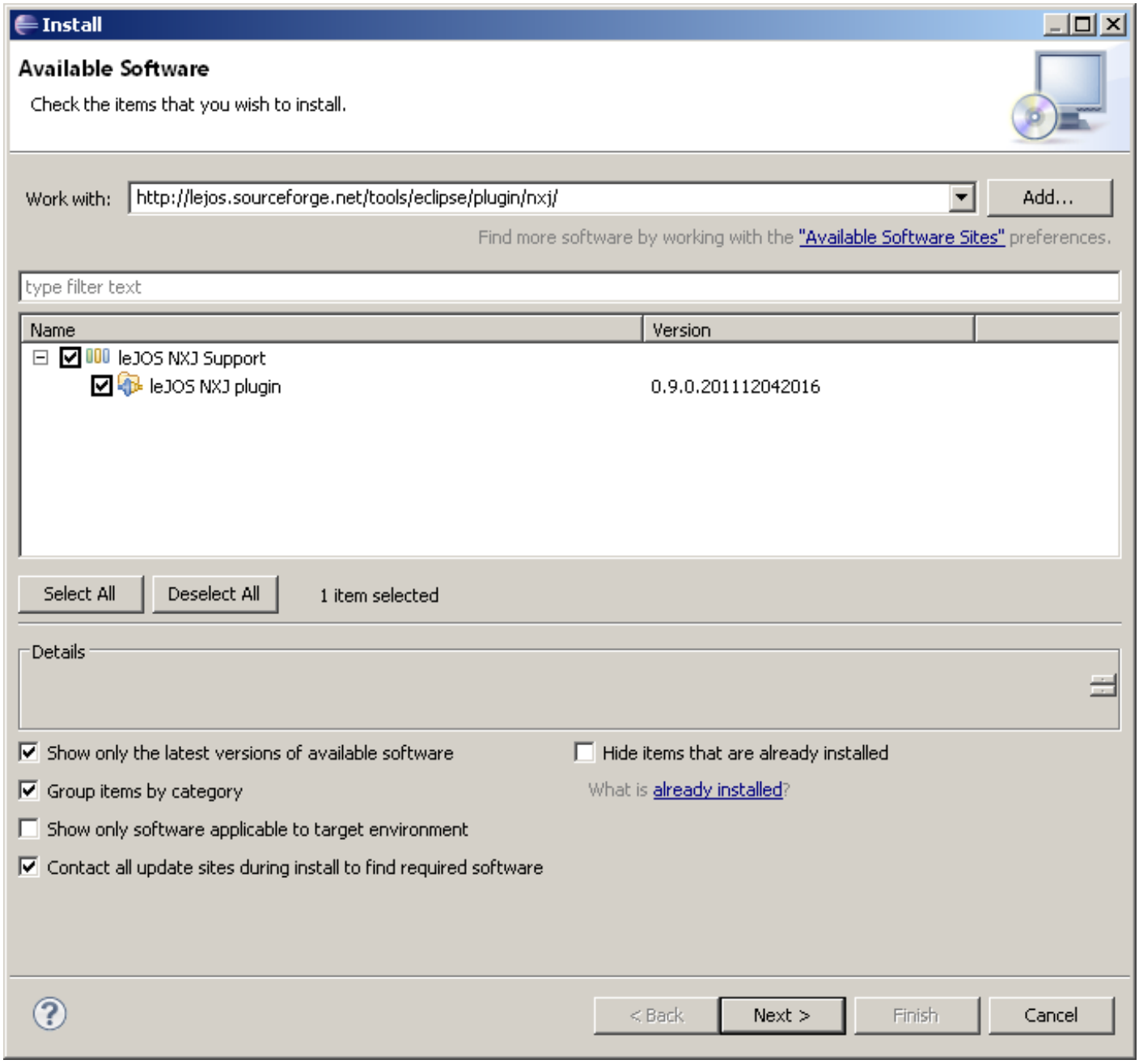
Installing the Eclipse plugin

The preferred way of setting up a project for leJOS with Eclipse is to use the leJOS Eclipse plugin. The plugin will allow you to create NXT projects for writing Java programs that run on the NXT itself and PC projects for writing Java program that remote control the NXT via USB or Bluetooth.

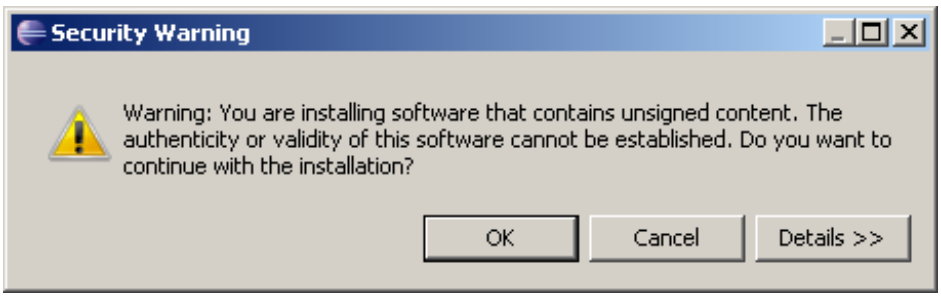
To install the leJOS, click on the Help menu and select "Install New Software..." as shown below.



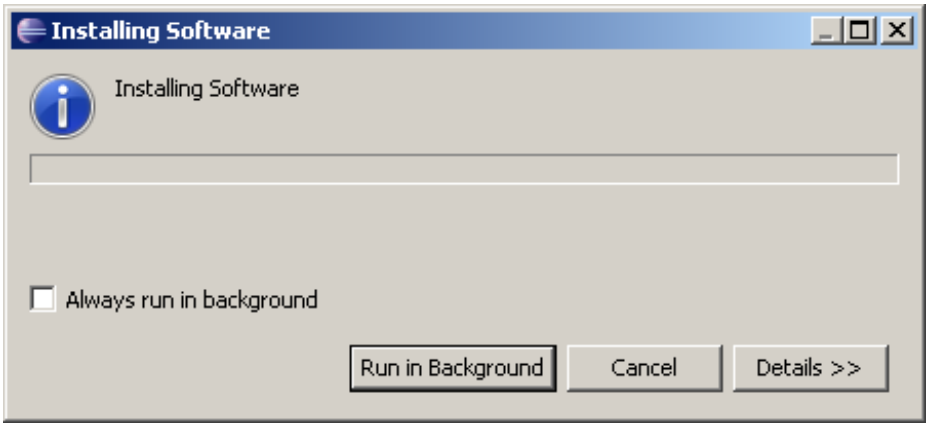
When you see the following dialog you type "http://lejos.sourceforge.net/tools/eclipse/plugin/nxj/" in the field labeled "Work with:". Then hit enter. Eclipse will check the update site and show a list of available components:



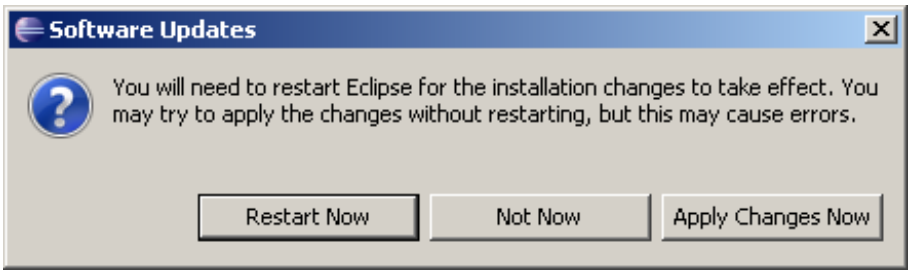
Chose to install the leJOS plug-in and click the Next button. Click through the next few pages of the Install dialog and eventually click on Finish. Eclipse will now download the plug-in. The leJOS plug-in is not signed and the following warning will appear:



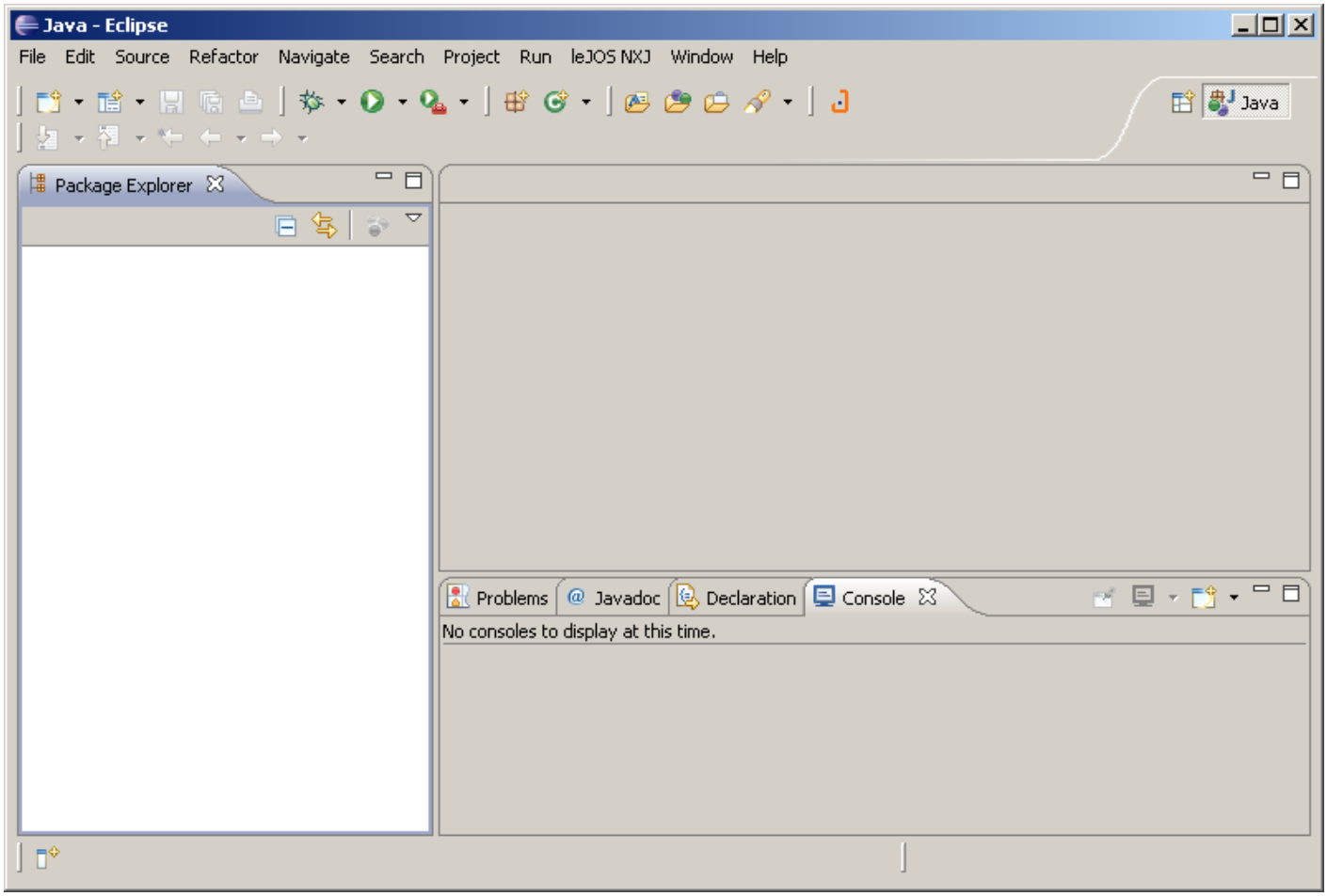
Click on OK and Eclipse will continue with installing the plug-in:



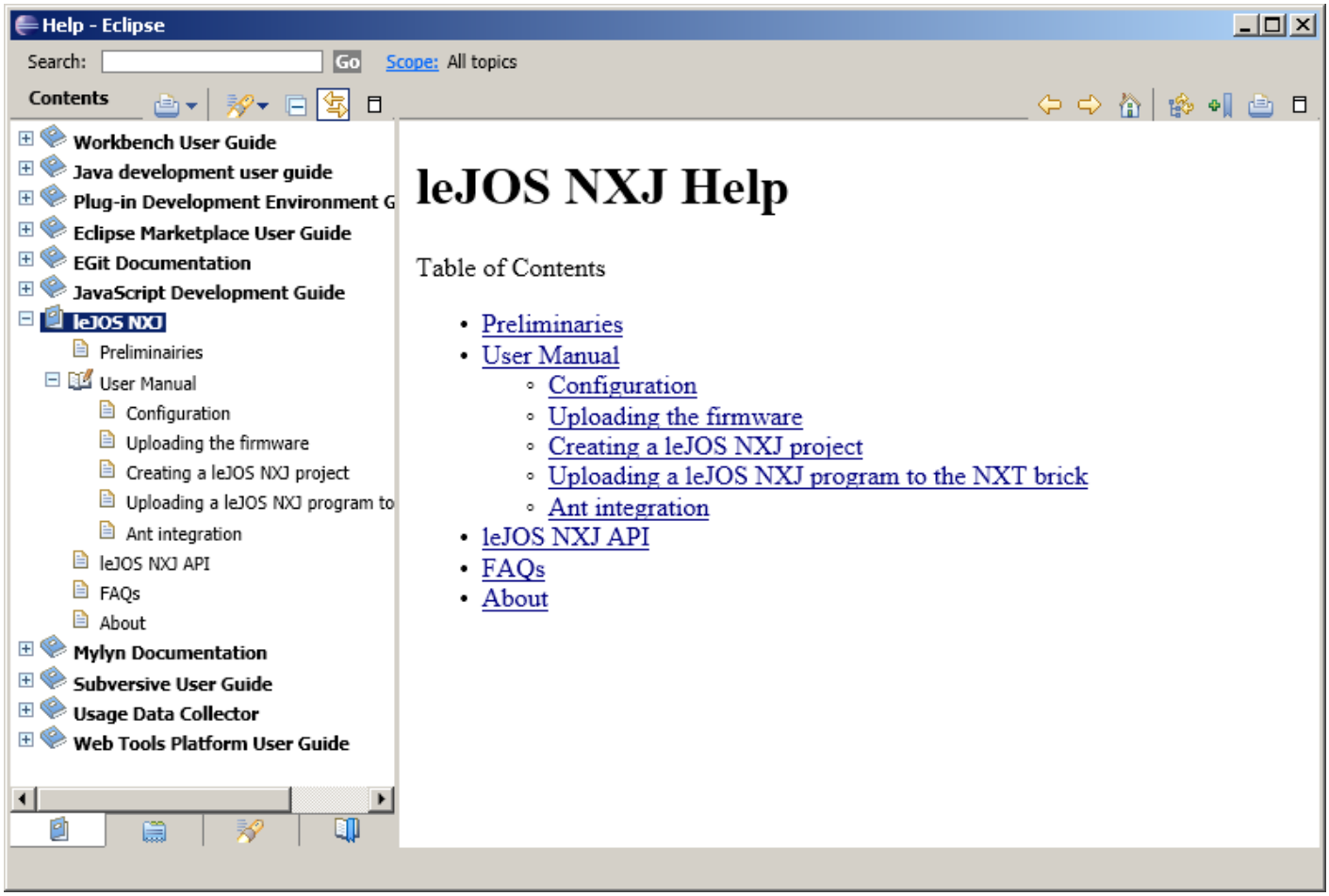
Wait until you are prompted to restart Eclipse:



Click on "Restart Now" and Eclipse will restart.

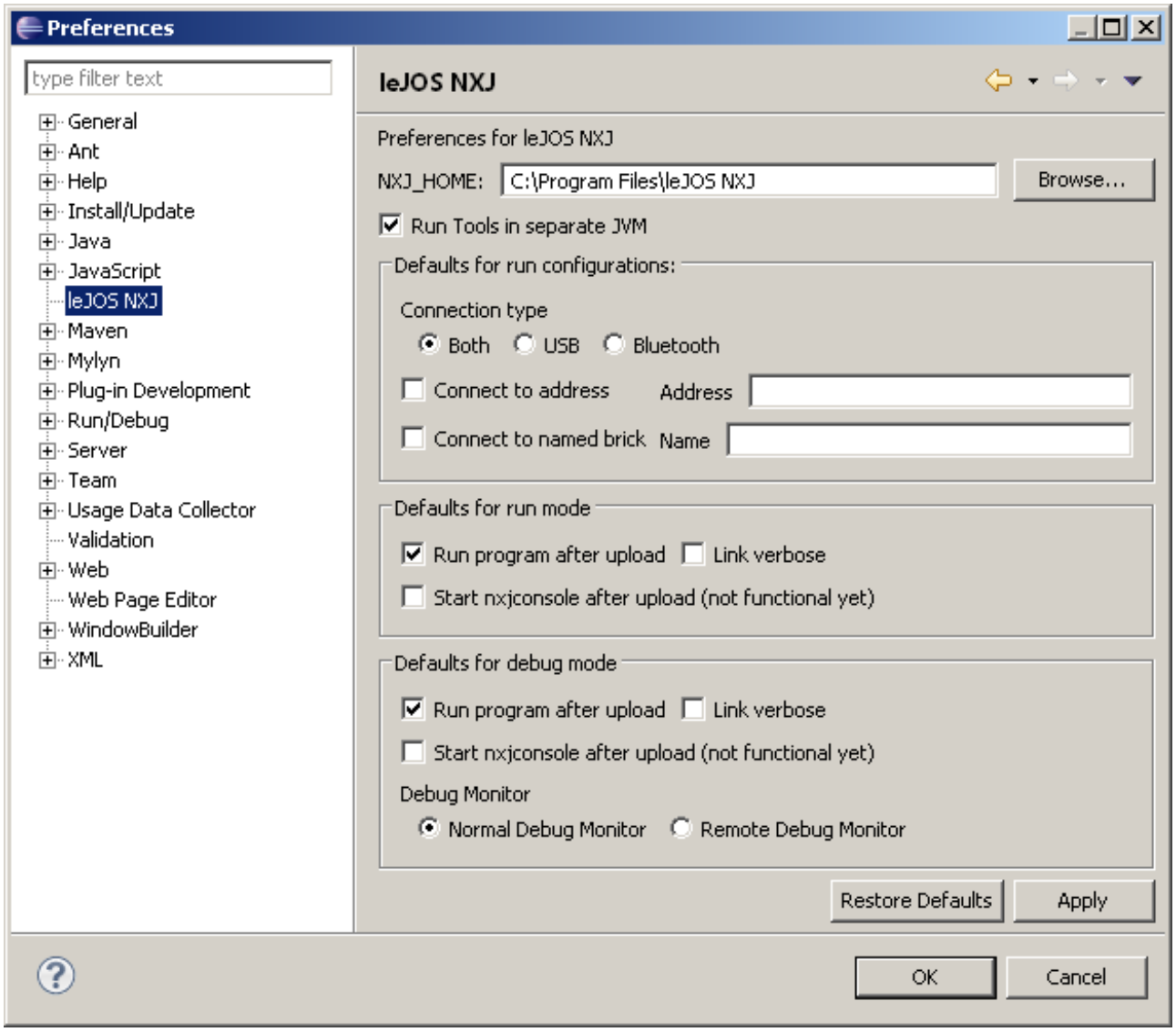


Notice the new button with the leJOS icon in the toolbar. You can read the plugin help by select clicking on the Help menu and selecting "Help Contents and then "leJOS NXJ".



It is a good idea to read the help page to familiarize yourself with the plugin.

To Configure the plugin for your system and preferences, click on the "Window" menu and select Preferences and then "leJOS NXJ".

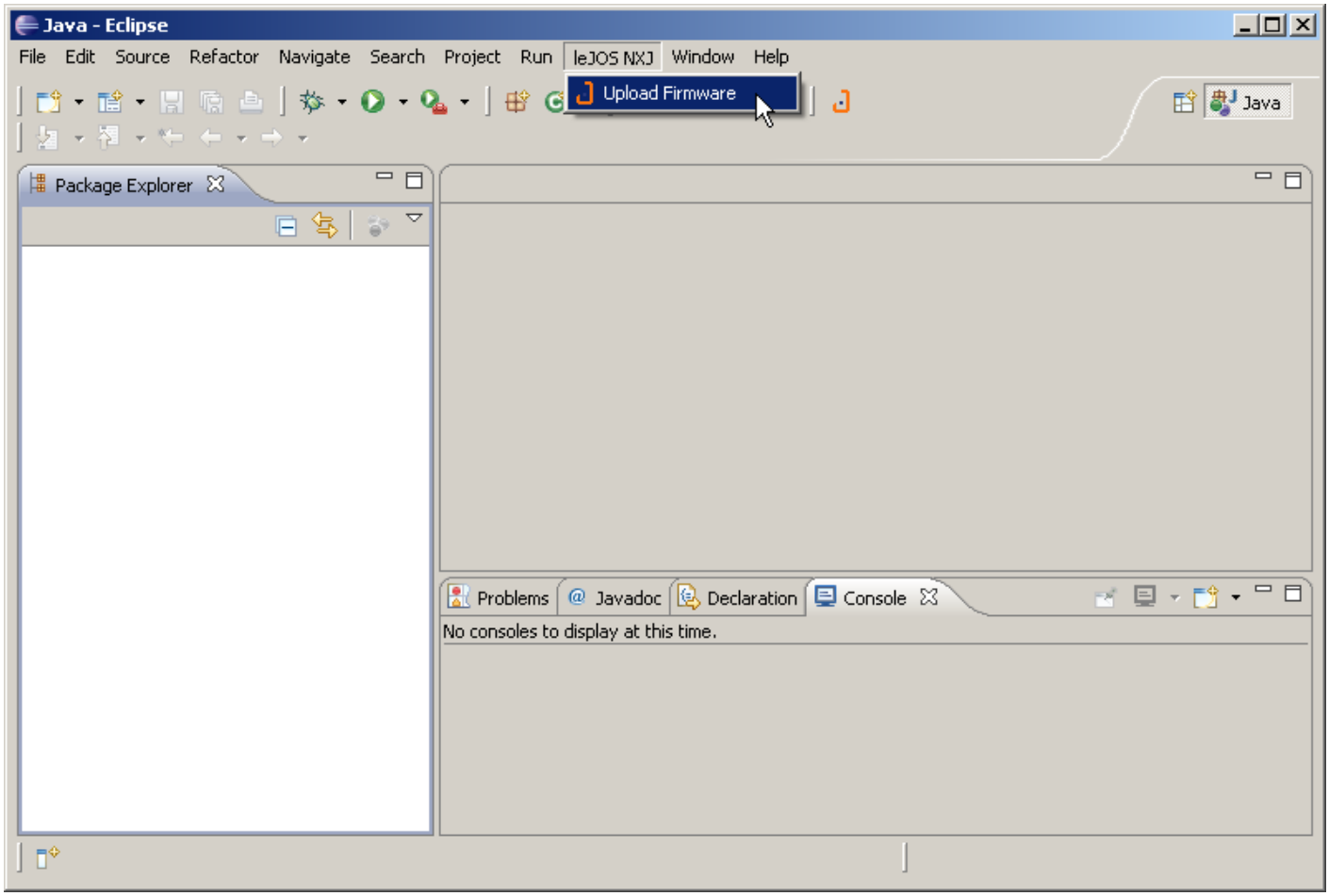


Browse to where you installed leJOS NXJ and select it as NXJ_HOME. Select any other options you require. It is a good idea to check the "Verbose" option. Note, that these setting are the *defaults* for the created Run Configurations. See below on how to edit them.

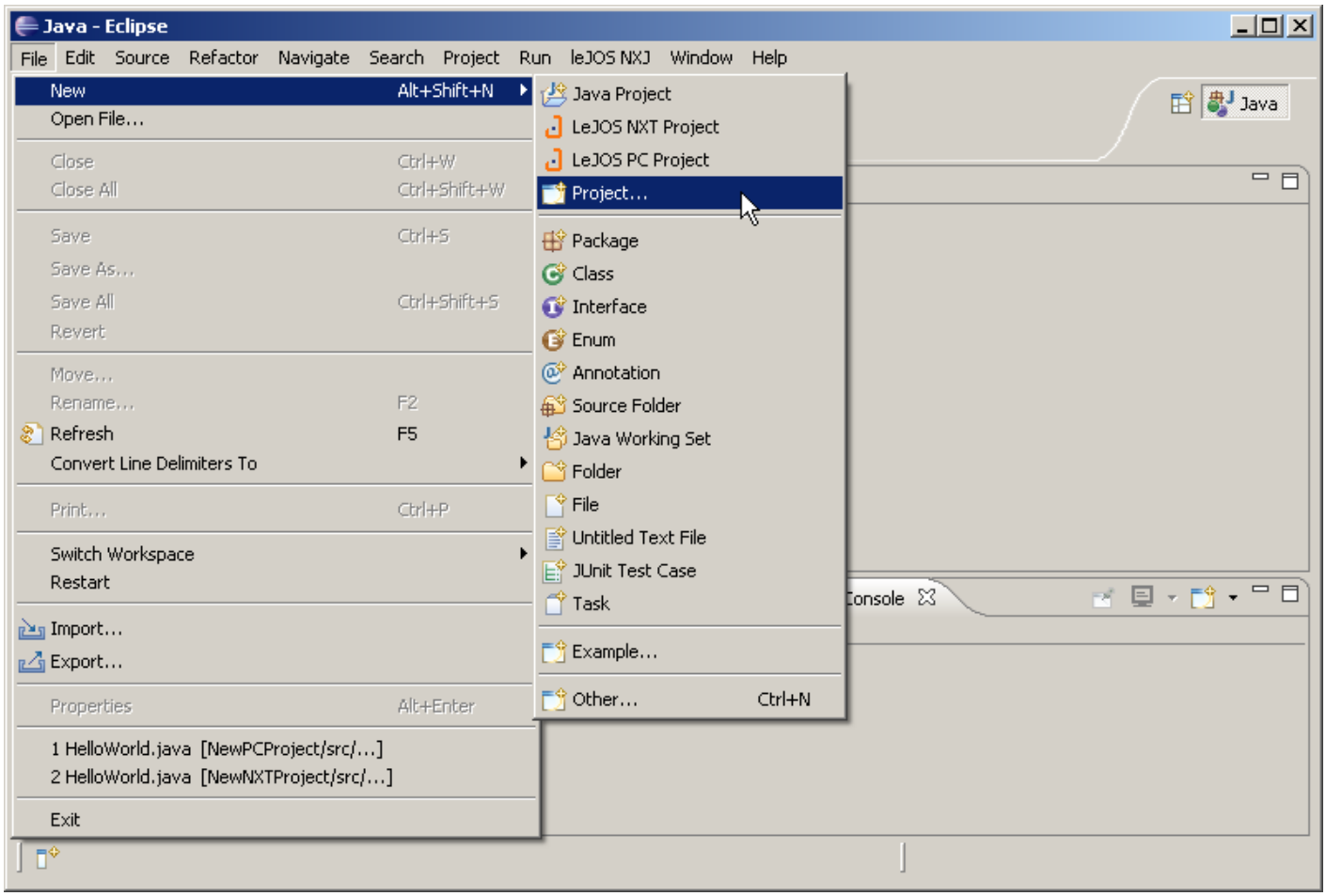
When you have set your preferences, click on "Apply" and then "OK".

The plugin is now set up and ready to use.

You can upload the leJOS NXJ firmware to your NXT brick from the plugin, by clicking on the "leJOS NXJ" menu item and selecting "Upload Firmware" or by clicking on the leJOS button on the toolbar. That will start the NXJFlash utility.

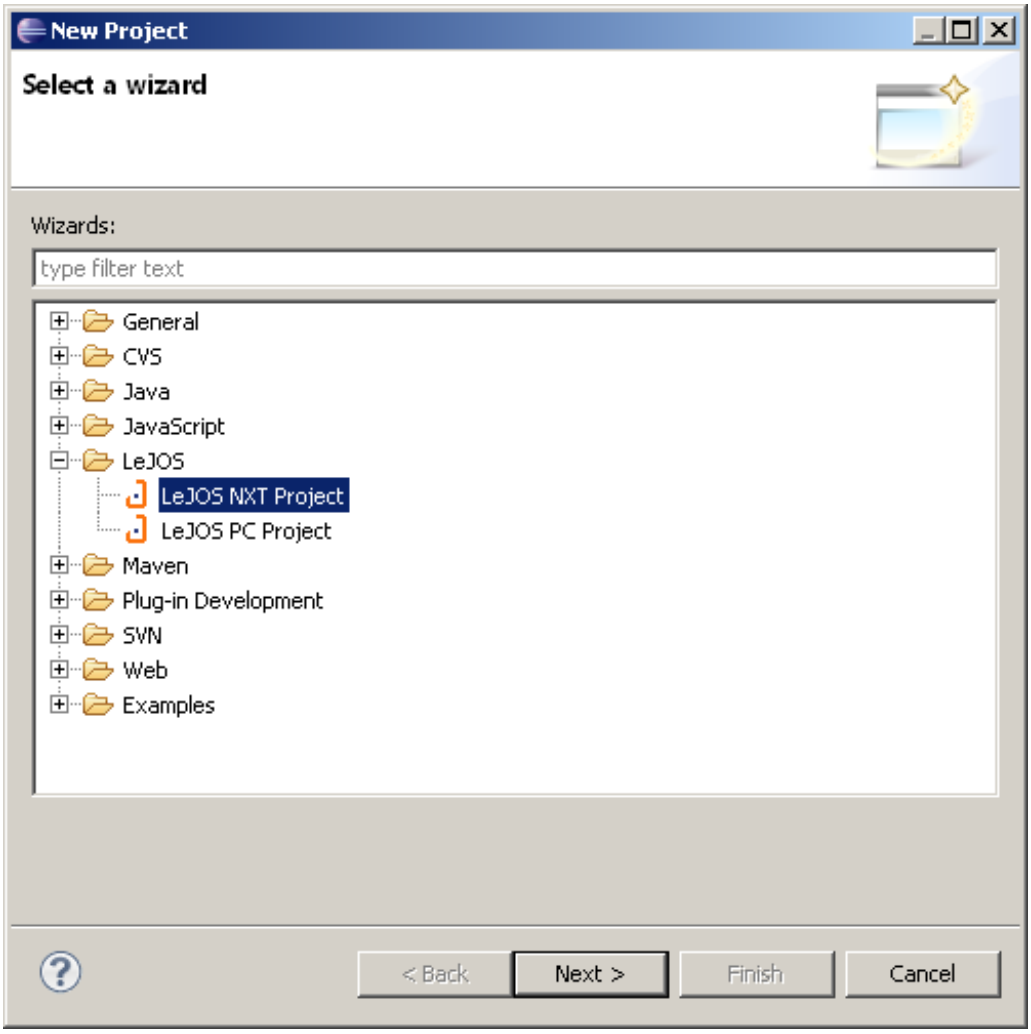


To create a new leJOS NXJ project using the plugin, go the "File" menu and select "New" and then "Project..."" as shown below:

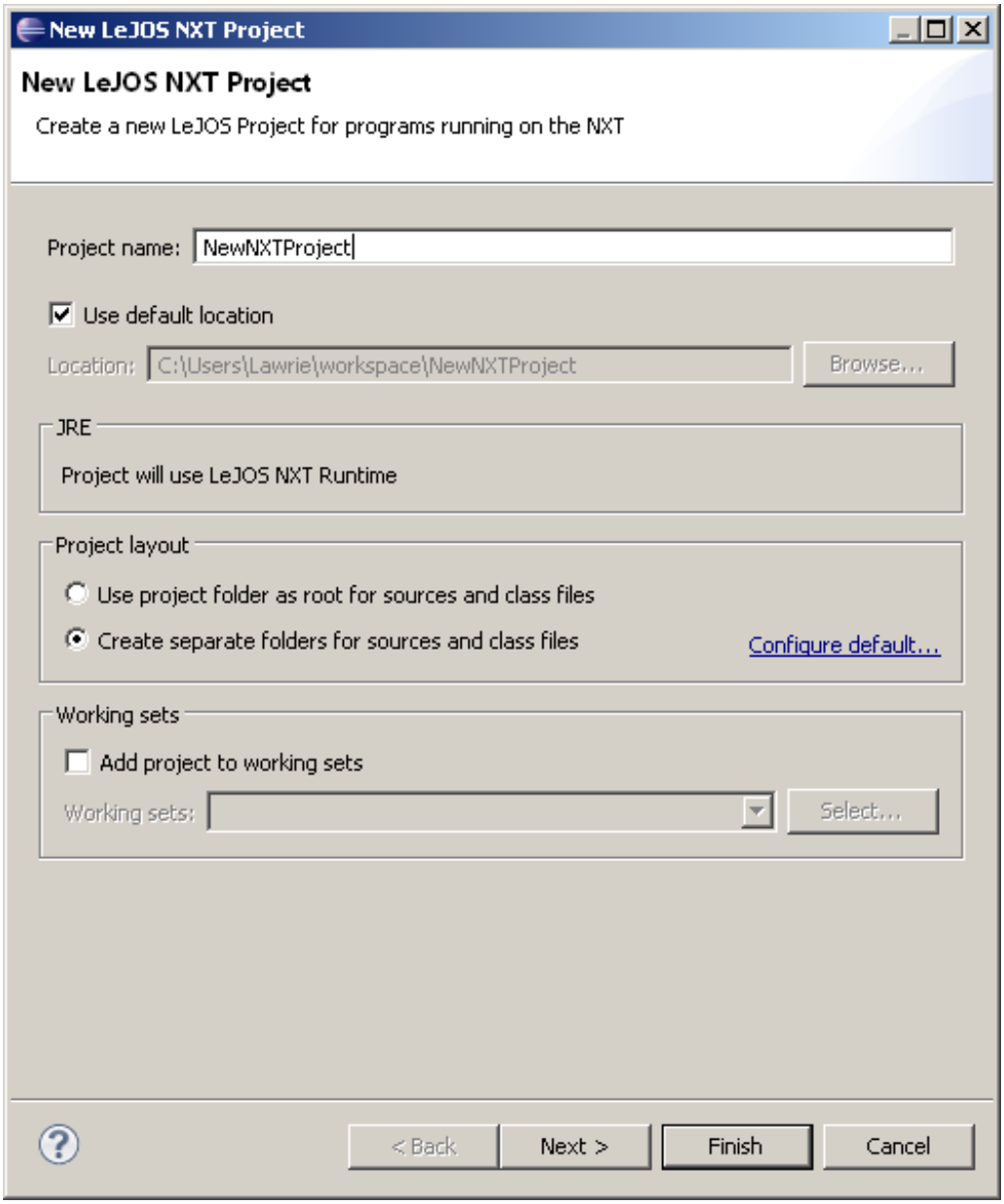


Creating your own NXT project

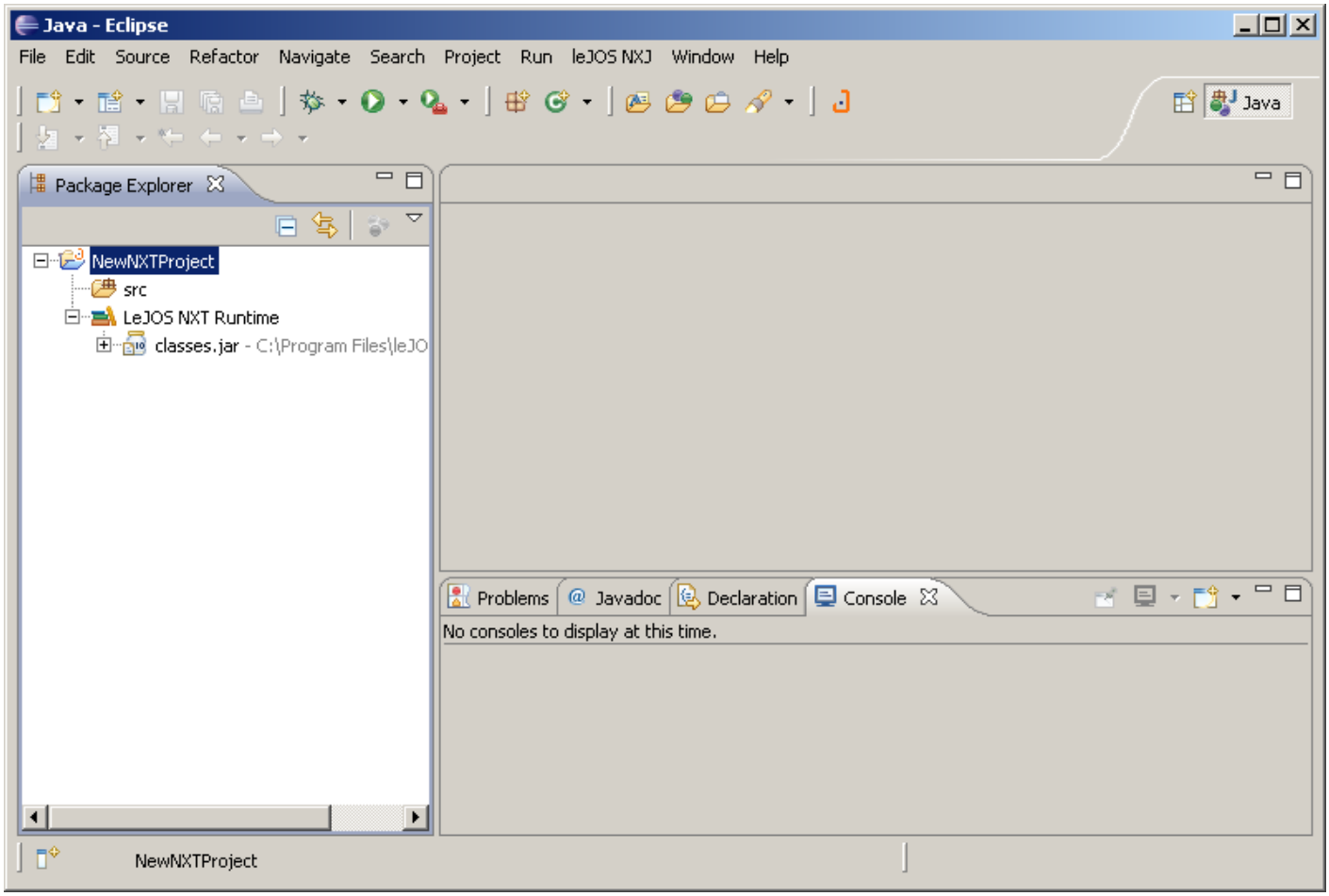
Open the new project wizard as explained above. Unfold the LeJOS category and select "LeJOS NXT Project" as shown below:



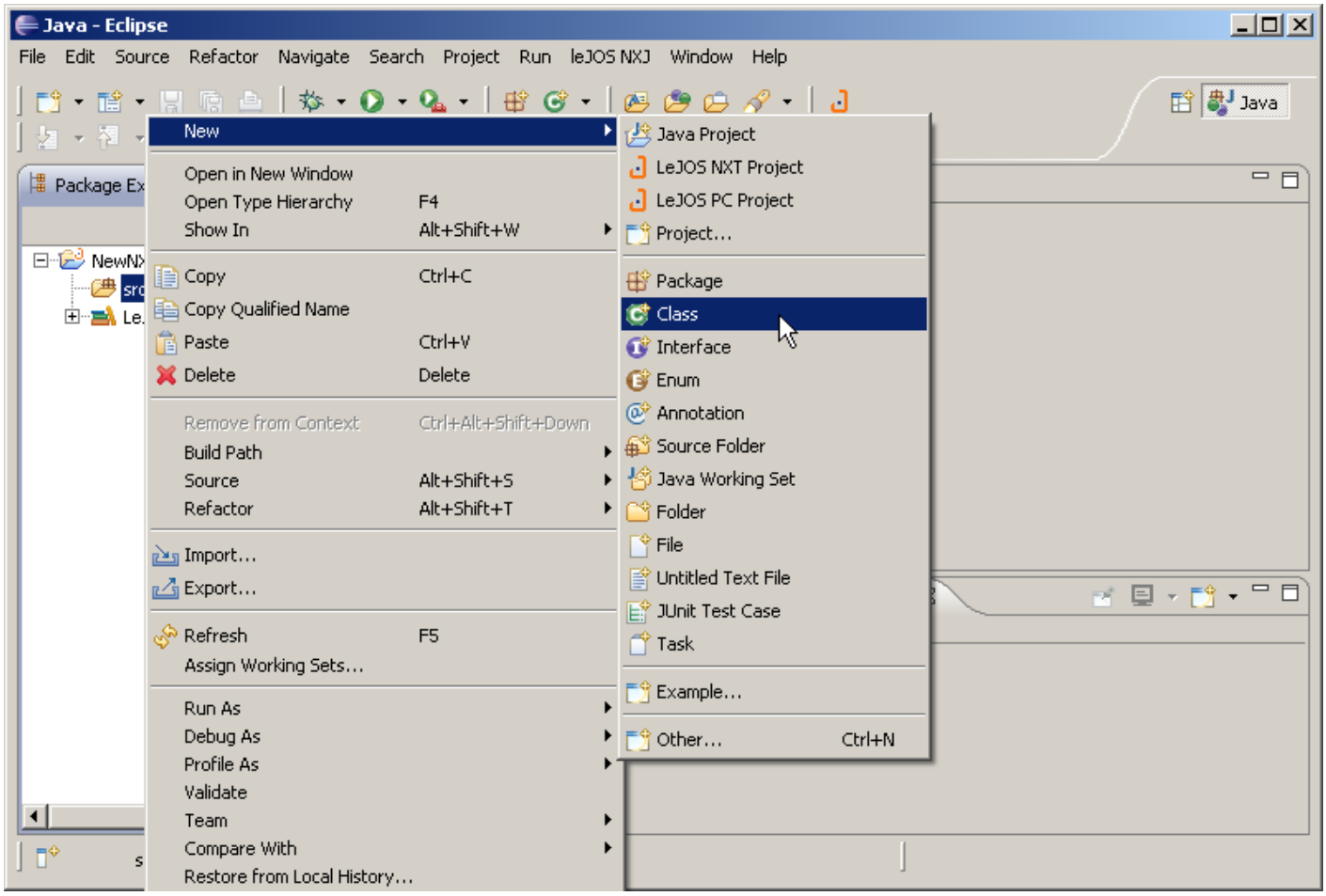
This will open the wizard for creating new leJOS NXT projects:



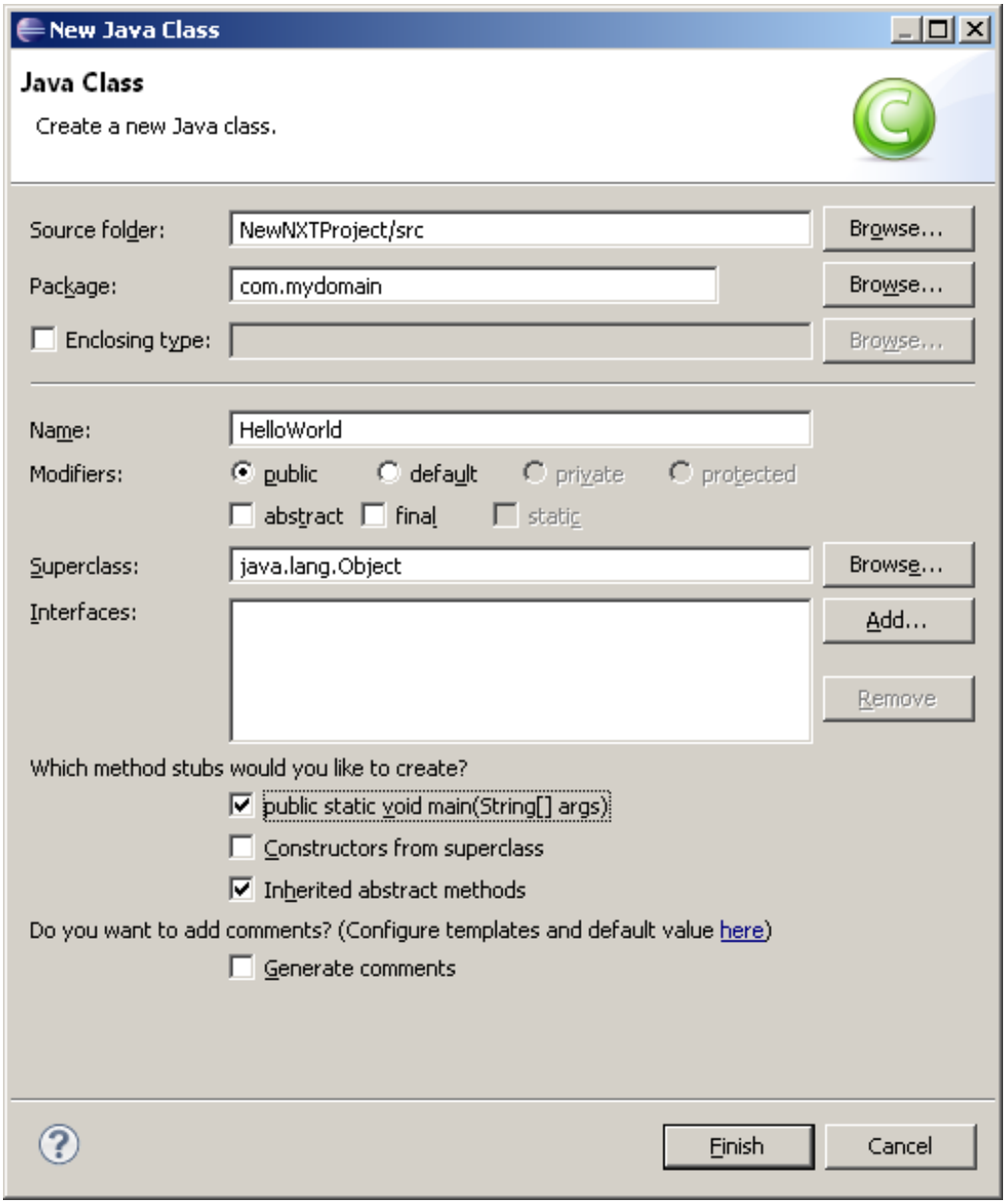
Enter the name of the project and click Finish. Back in the main window of Eclipse, you will see your new project. Note, that the classes.jar (which replaces the Java Runtime) is already included in the projects build path. Also note the small leJOS icon next to the project's name. This indicates that the project is for NXT programs only.



Now right click the src folder and select "New" and then "Class" in the context menu as shown below:



This will open the wizard for creating new Java classes:



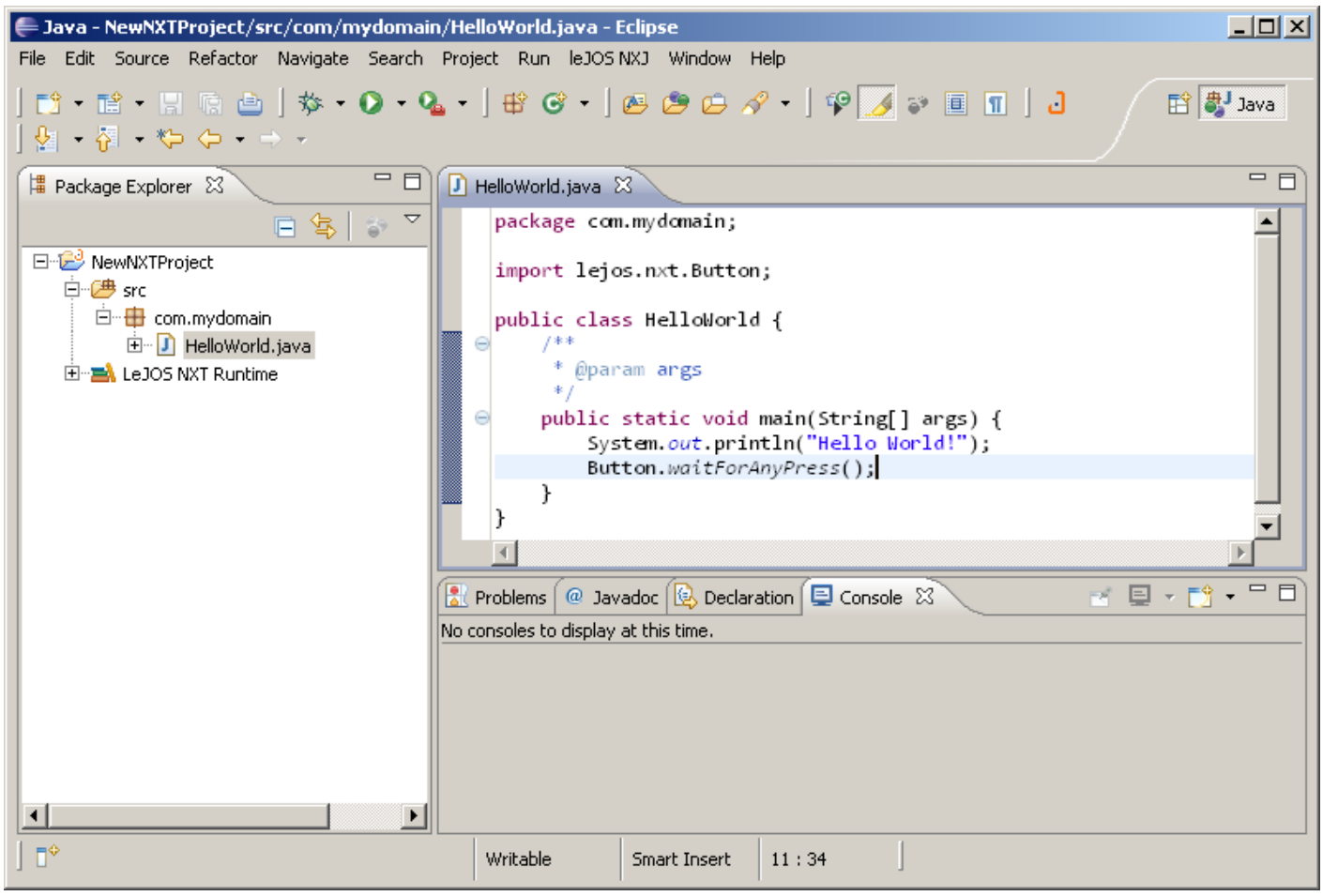
Enter a package name and the a the name of the class. In the example, the class name HelloWorld has been chosen. The package name `com.mydomain` corresponds to the DNS domain "mydomain.com".

Make sure to chose appropriate names. Typically, the package name consist of a DNS domain name in reverse order followed by the name of the project, for example.

Once you click Finish, the editor for the class will open. Now you can start programming. The example in the screenshot shows the following code:

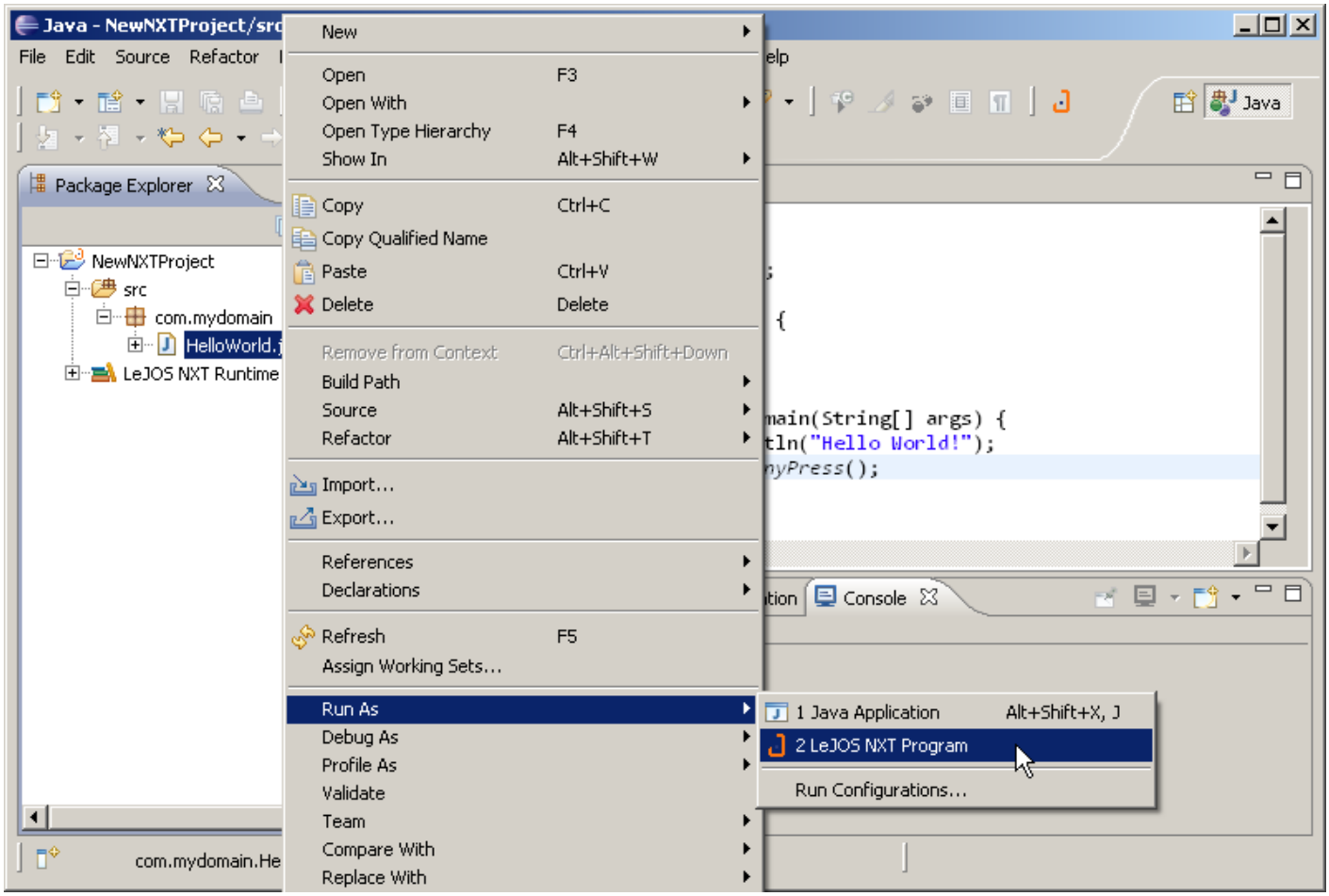
```
package com.mydomain;

import lejos.nxt.Button;

public class HelloWorld {
    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        Button.waitForAnyPress();
    }
}
```

The program will print "Hello World!" on the NXT's LCD screen and wait for any Button to be pressed. Of course you can add further classes and packages to your project.

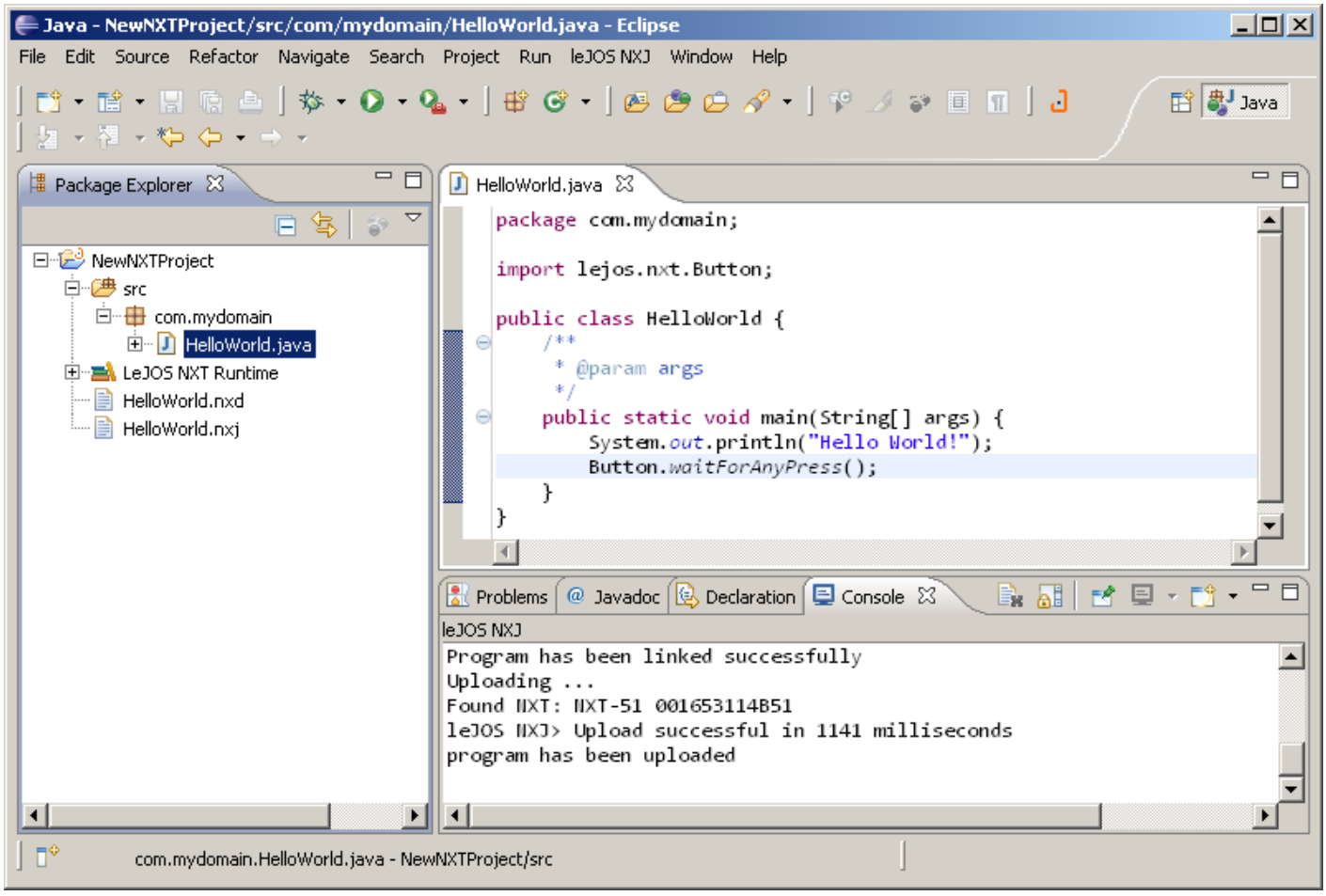


When you are ready to upload your program to the brick, right-click on the class which contains the `main` method. Select "Run As"→"LeJOS NXT Program" in the context menu.

Note: do not select "Run As"→"Java Application" since this will try start the program locally, with a normal Java virtual machine and a normal Java runtime. It will not work and almost certainly result in an error.



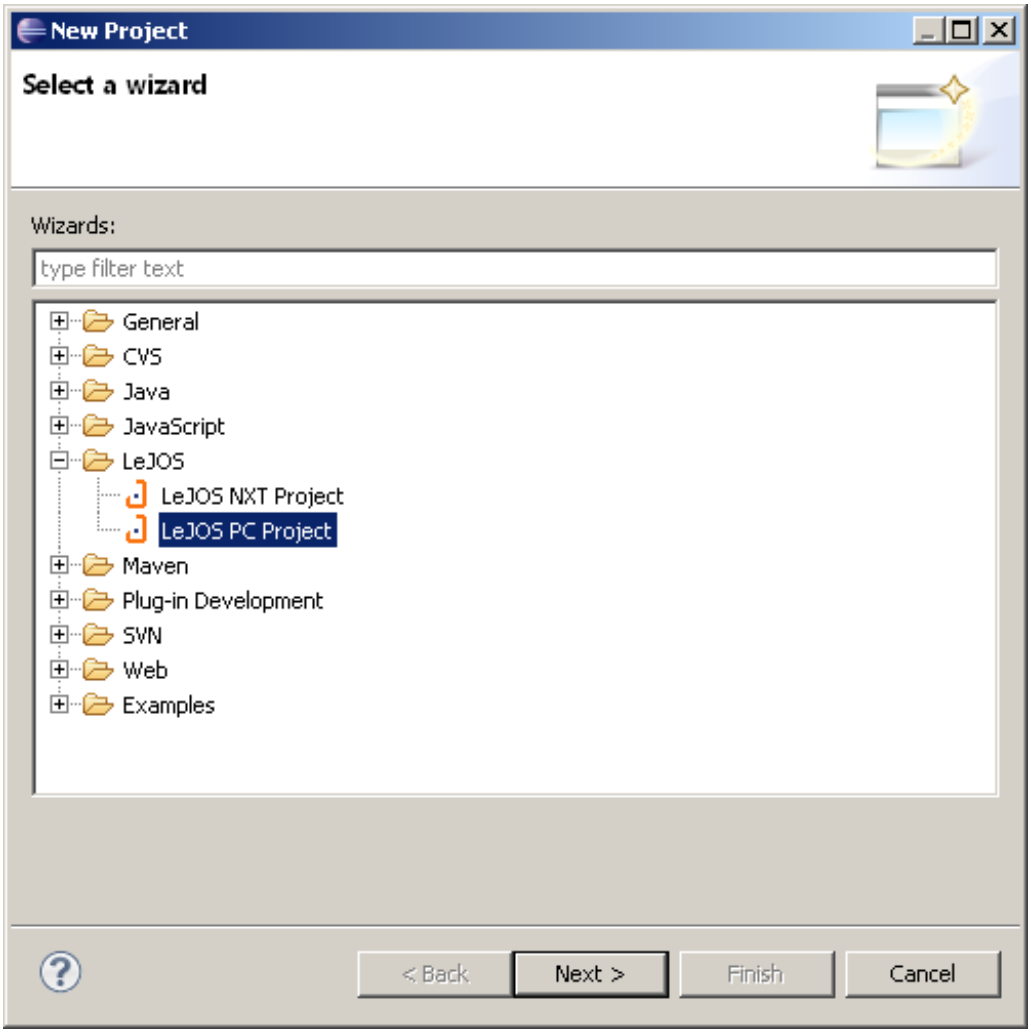
You will see output on a leJOS NXJ Console. Also note, that two additional files have been created in the project root directory. The nxj file is the one being upload to the NXT brick. The nxd is for debugging purposes.



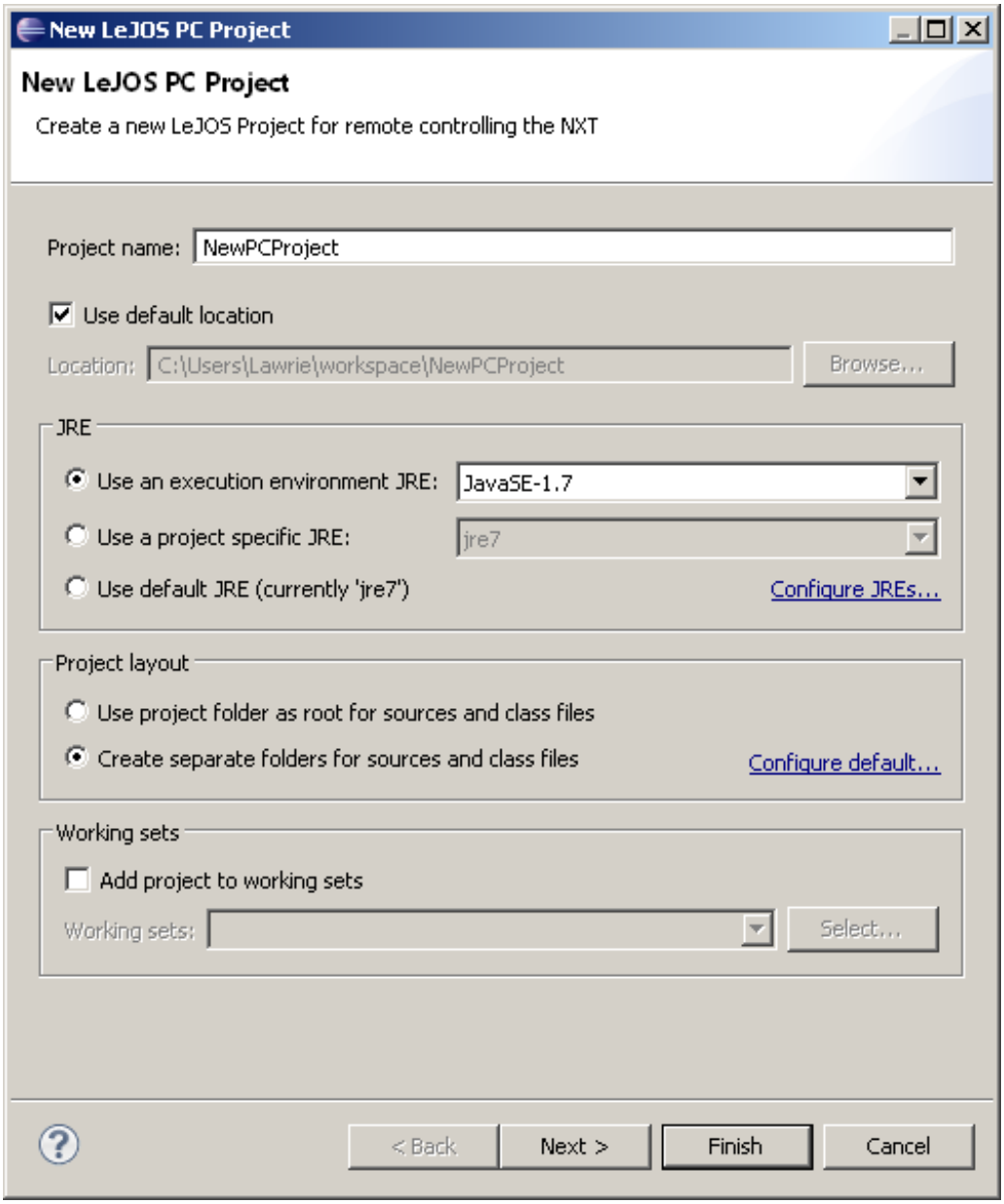
If you selected the "Run program after upload" option, the program will start running on your NXT when it has finished uploaing.

Creating your own PC project

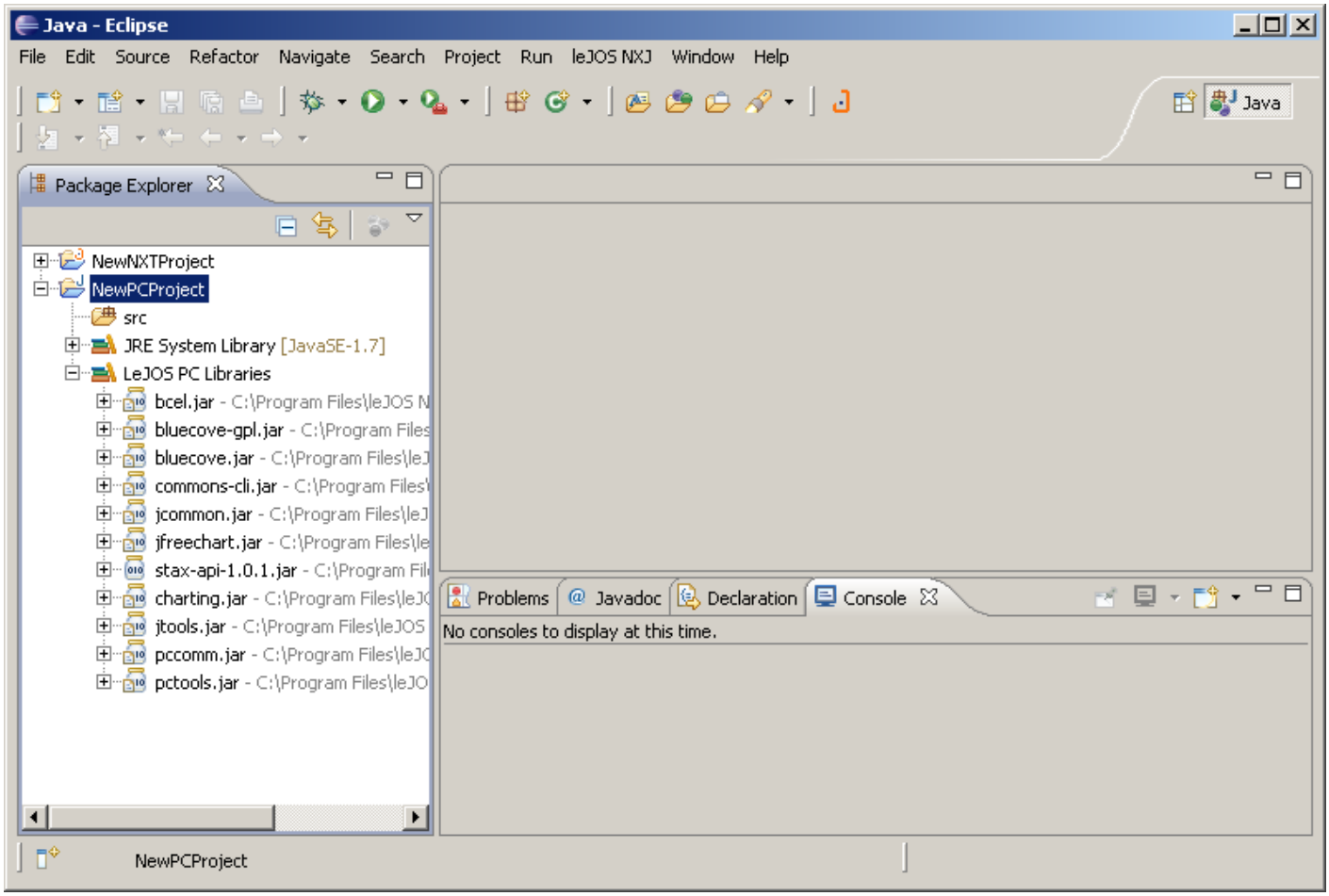
Open the new project wizard as explained above. Unfold the LeJOS category and select "LeJOS PC Project" as shown below:



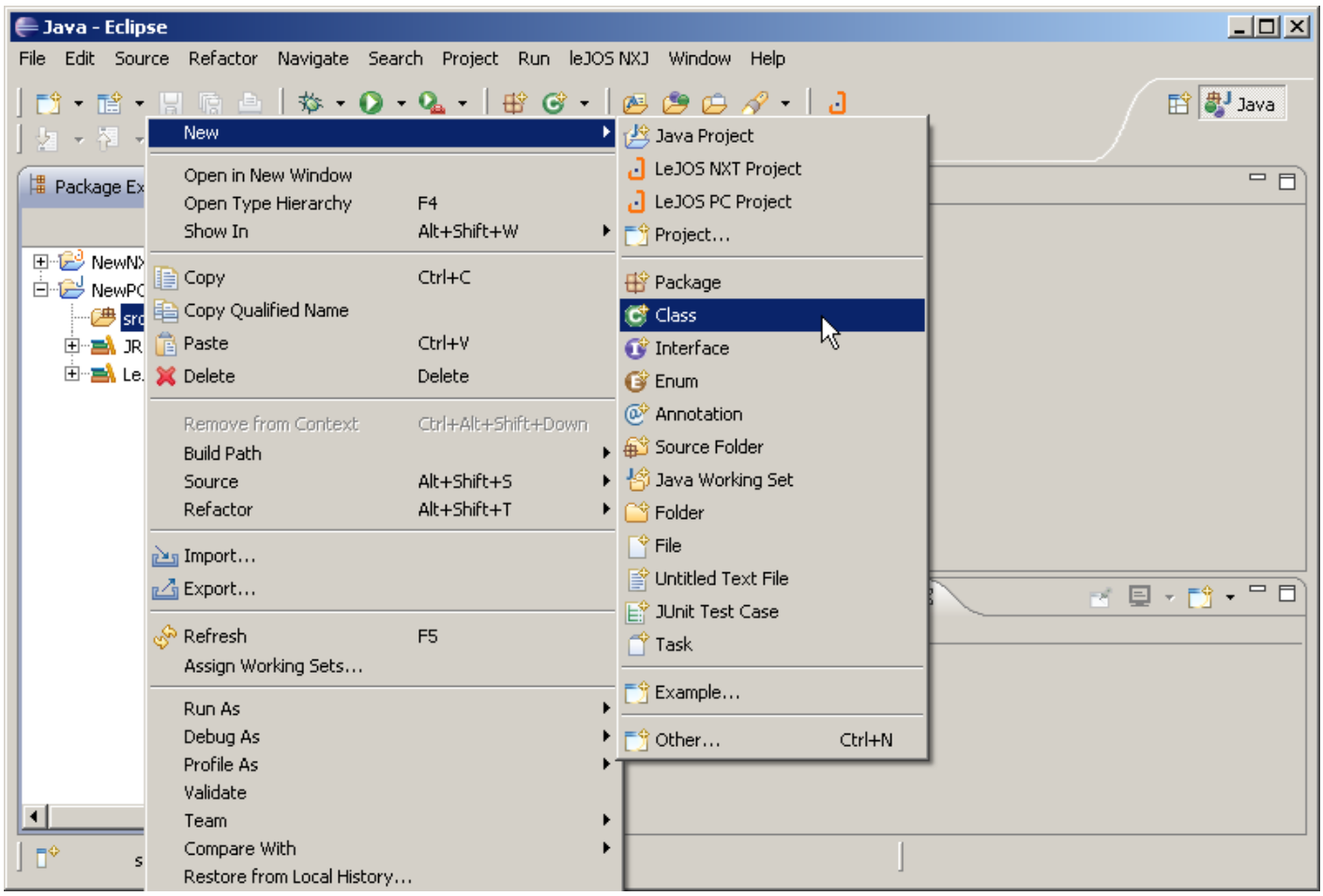
This will open the wizard for creating new leJOS PC projects:



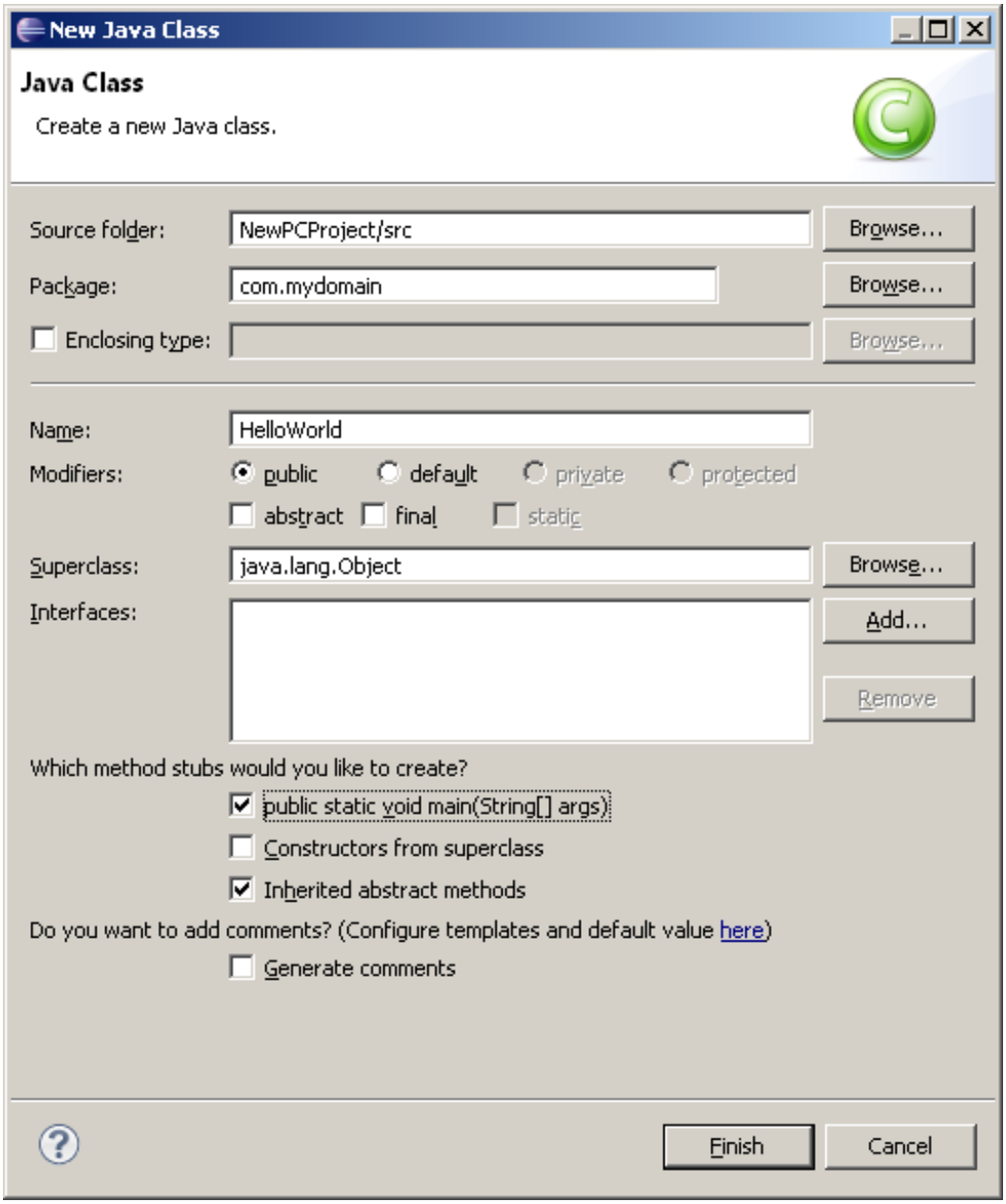
Enter the name of the project and click Finish. Back in the main window of Eclipse, you will see your new project. Note, that the JAR files like pccomm.jar etc. and their dependencies are already included in the projects build path. Note, that PC projects don't have the small leJOS icon next to the projects name.



Now right click the src folder and select "New" and then "Class" in the context menu as shown below:



This will open the wizard for creating new Java classes:



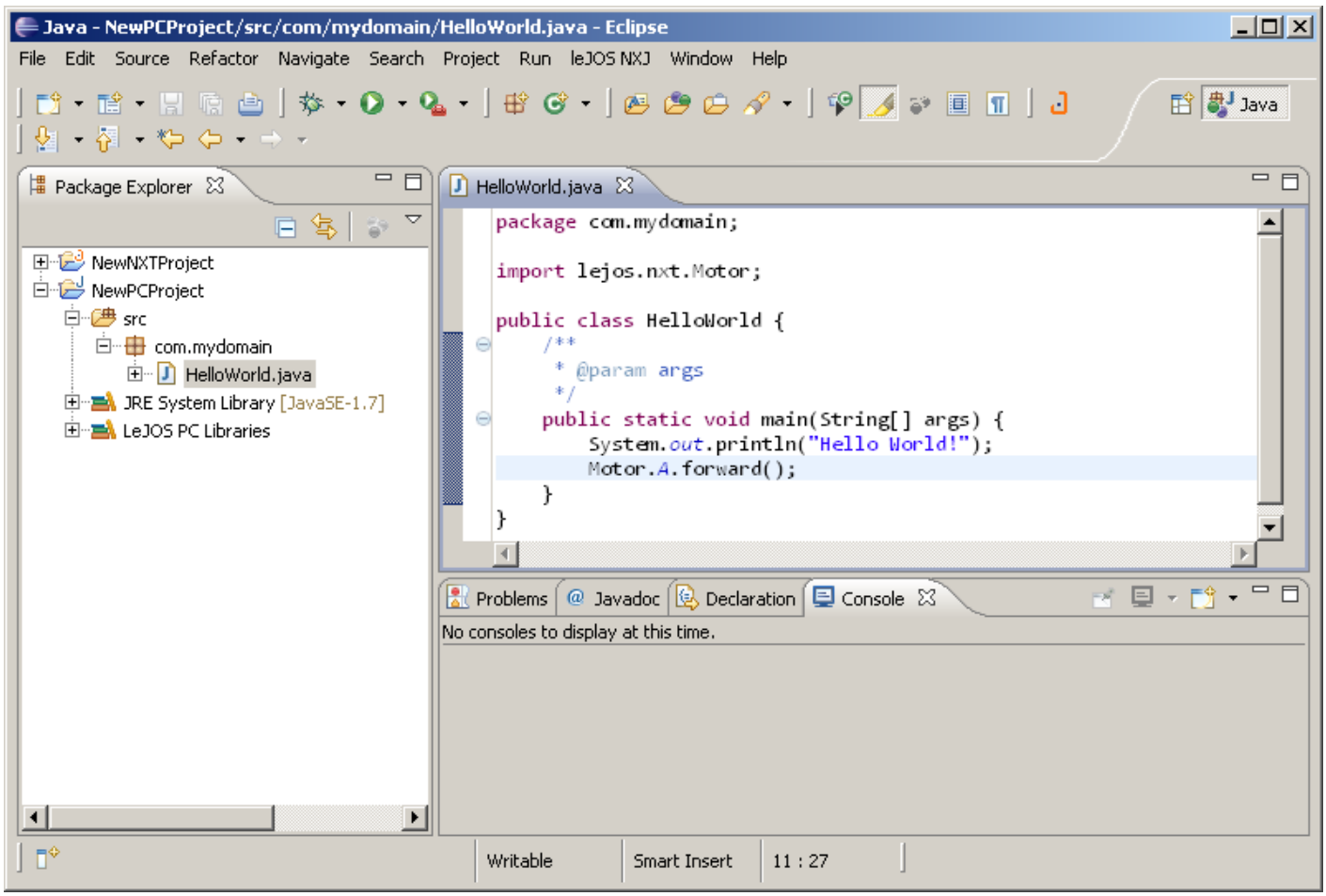
Enter a package name and the a the name of the class. In the example, the class name HelloWorld has been chosen. The package name `com.mydomain` corresponds to the DNS domain "mydomain.com".

Make sure to chose appropriate names. Typically, the package name consist of a DNS domain name in reverse order followed by the name of the project, for example.

Once you click Finish, the editor for the class will open. Now you can start programming. The example in the screenshot shows the following code:

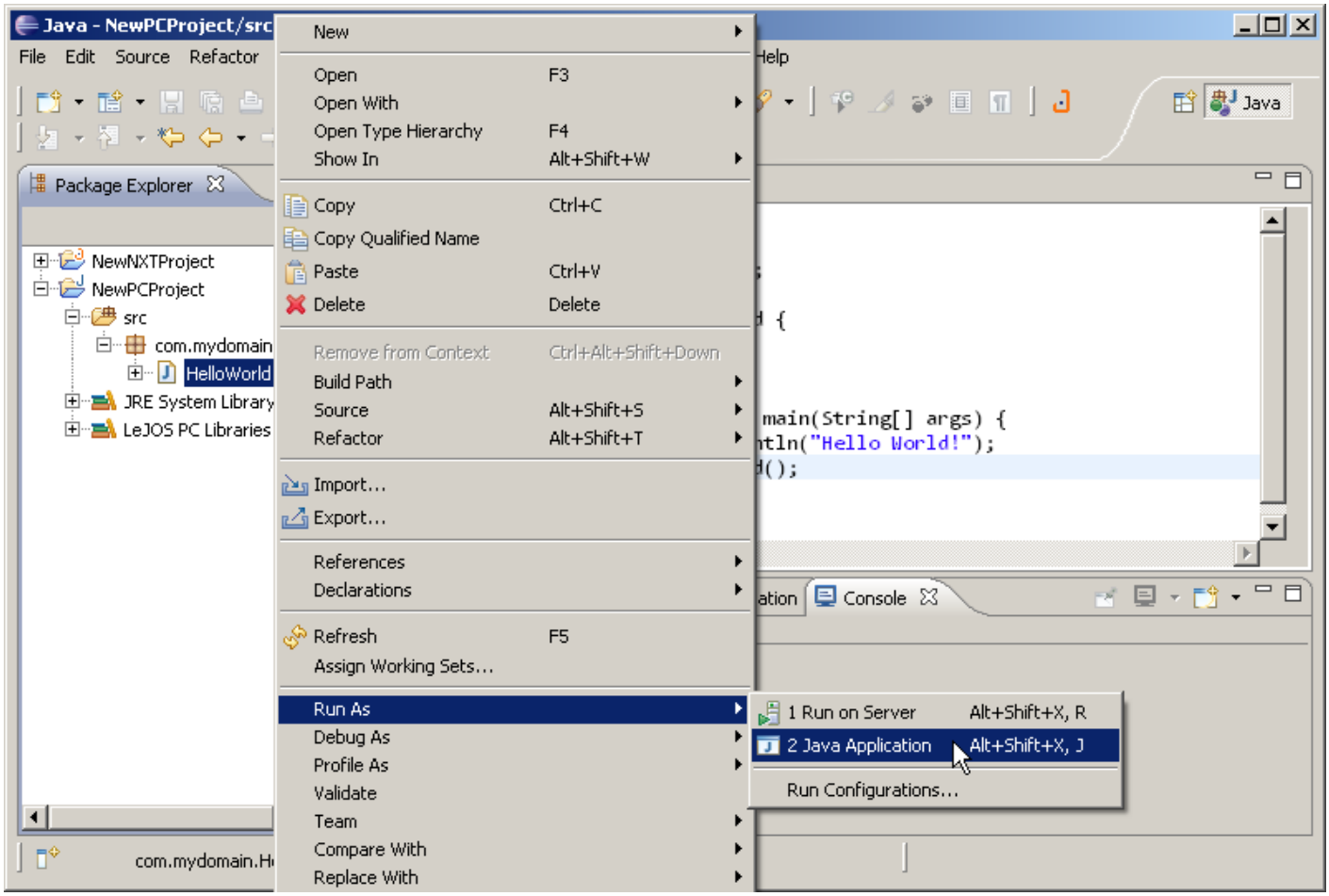
```
package com.mydomain;

import lejos.nxt.Motor;

public class HelloWorld {
    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        Motor.A.forward();
    }
}
```

The program will print "Hello World!" locall on the PC. Then it will try to connect to any NXT and will start the Motor connected to Port A. Of course you can add further classes and packages to your project.

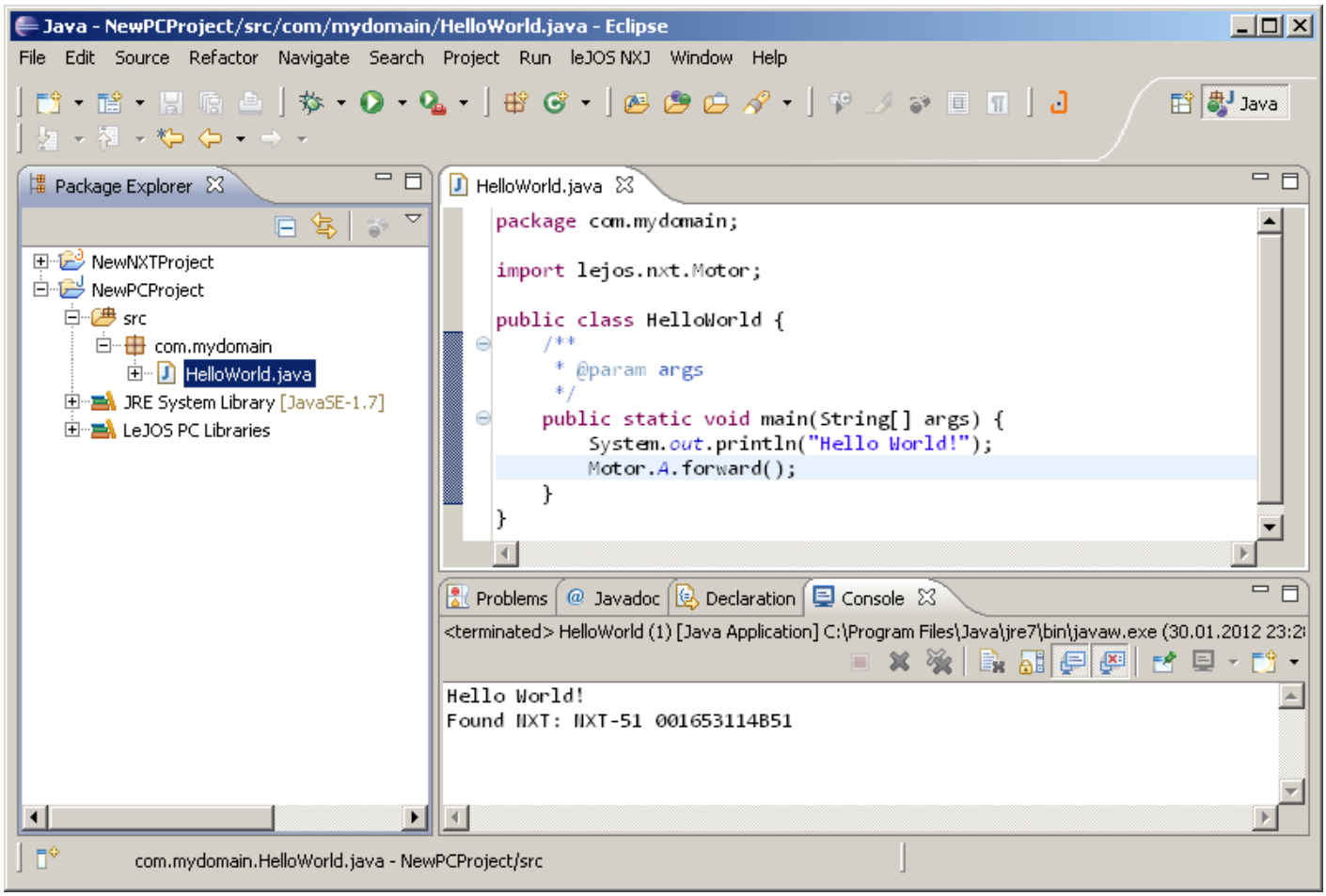


When you are ready to run your program, right-click on the class which contains the `main` method.

Select "Run As"→"Java Application" in the context menu.

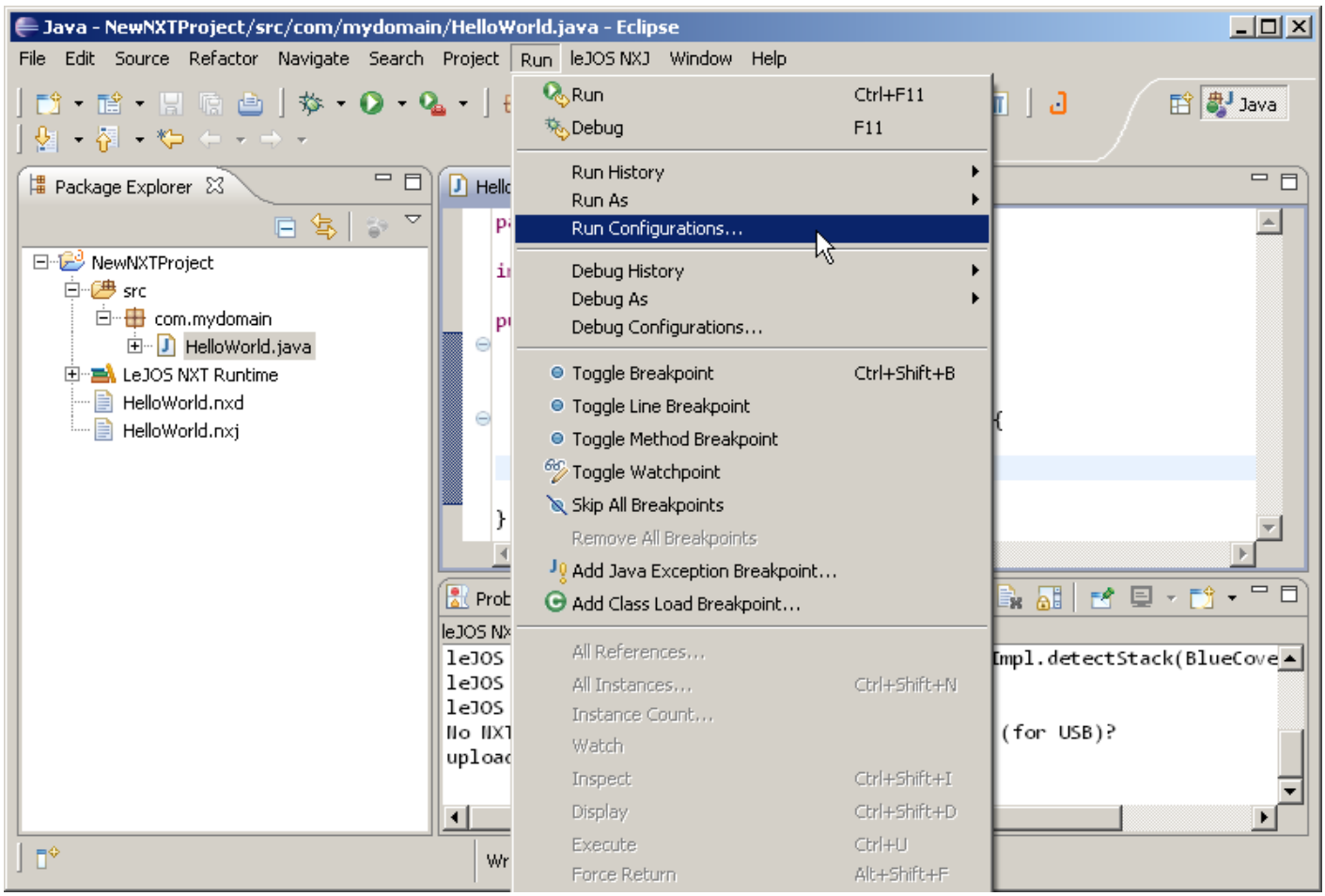


You will see output on the local console in Eclipse. Note the line after the "Hello World!" which indicates that the program connected to an NXT brick.

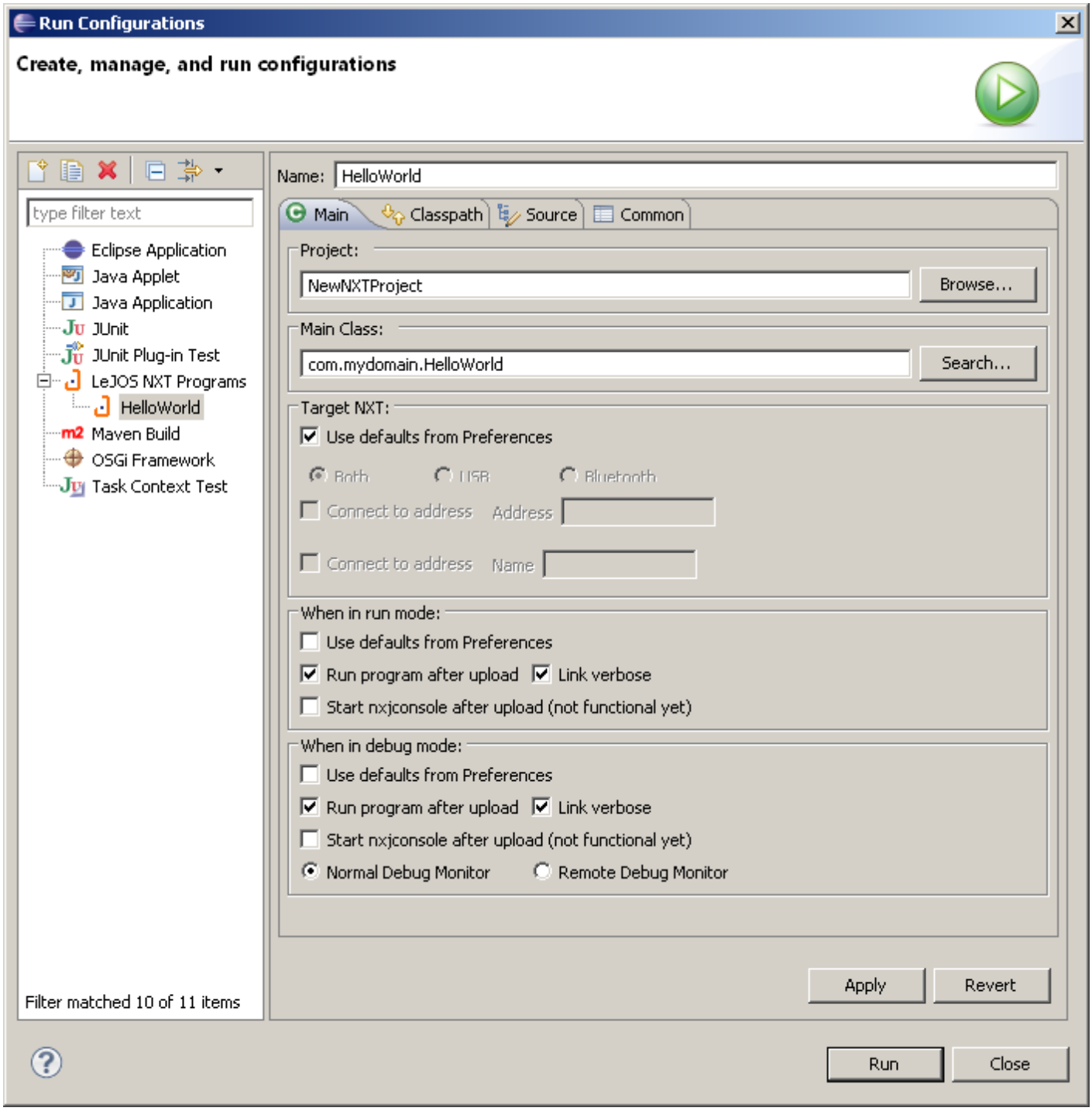


Editing Run Configurations

When chosing launching a LeJOS NXT Program or a Java Application in Eclipse, a so-called Run Configuration is created. It can be edited in order to change certain parameters. In the example below, the verbose linking is enabled for a specific Run Configuration. First select the "Run"→"Run Configurations..." in the menu:



The following dialog will open. Select the Run Configuration you want to change:



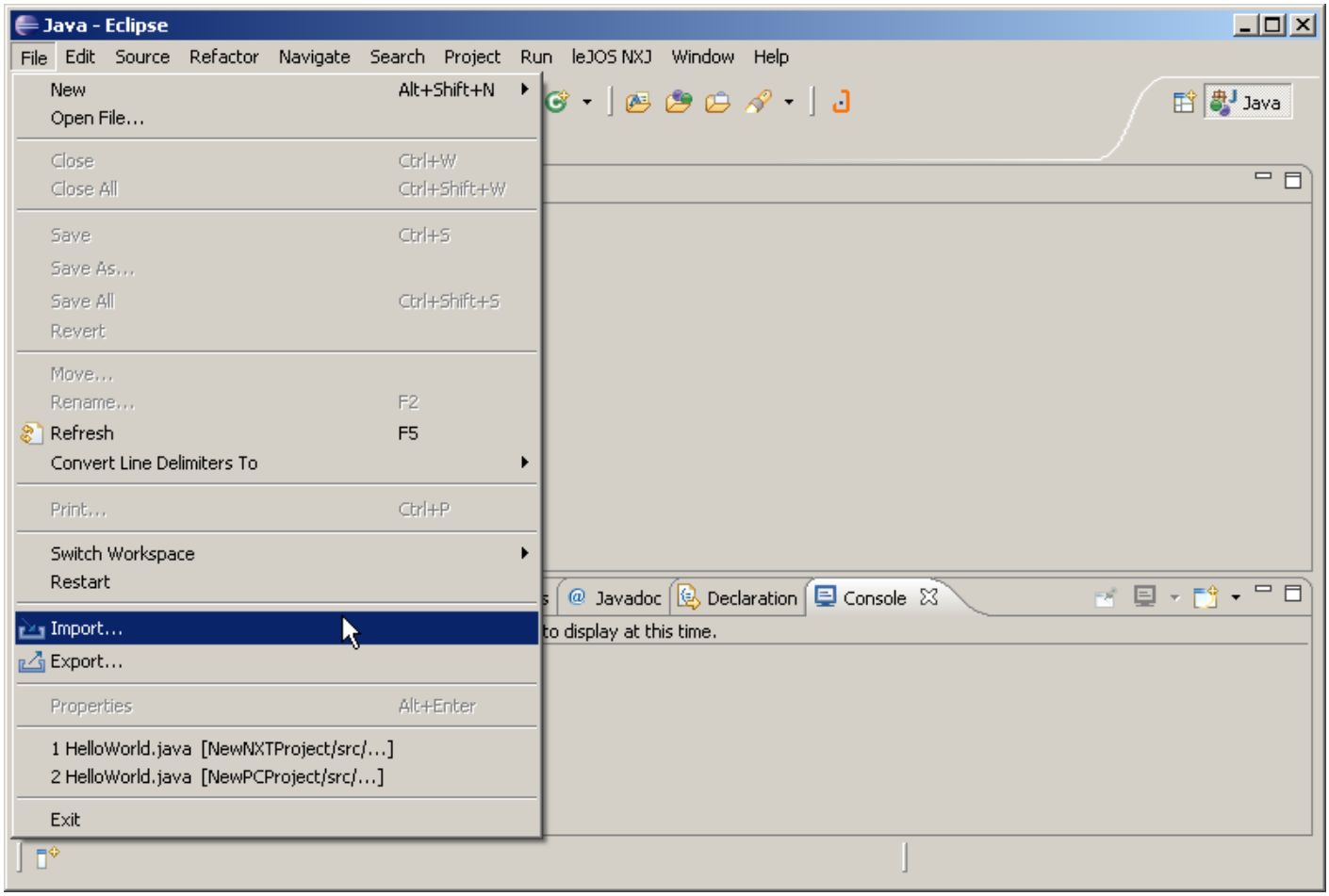
After making the changes, click "Apply" and then "Run" or "Close". After enabling the verbose linking, the output in the console window when uploading a program to the NXT will change:

```
Linking ...
Class 0: java.lang.Object
Class 1: java.lang.Throwable
Class 2: java.lang.Error
Class 3: java.lang.OutOfMemoryError
Class 4: boolean
Class 5: char
Class 6: float
...
Class 74: lejos.nxt.NXT
Method 0: java.lang.Object.<init>() PC 3880 Signature id 2
Method 1: java.lang.Object.getClass() PC 3881 Signature id 125
Method 2: java.lang.Object.toString() PC 3889 Signature id 127
Method 3: java.lang.Object.wait(long) Native id 8
Method 4: java.lang.Throwable.<init>() PC 3917 Signature id 2
Method 5: java.lang.Throwable.<init>(java.lang.String) PC 3927
Signature id 129
Method 6: java.lang.Throwable.getCause() PC 3942 Signature id 133
...
Method 163: lejos.nxt.NXT.getUserPages() Native id 93
Master record    : 20 bytes.
Class records    : 75 (900 bytes).
Field records    : 61 (64 bytes).
Static fields    : 38 (76 bytes).
Static state     : 38 (144 bytes).
Constant records : 40 (160 bytes).
Constant values  : 40 (524 bytes).
Method records   : 164 (1968 bytes).
Exception records: 20 (160 bytes).
Interface maps   : 3 (4 bytes).
Code             : 126 (3852 bytes).
Total            : 7732 bytes.
Run time options : EnableCompact
Constant loads   : 40N 8O 1W 39S
Static load/store: 6N 85O
Field  load/store: 2N 91O
Program has been linked successfully
Uploading ...
Found NXT: NXT-51 001653114B51
leJOS NXJ> Upload successful in 1342 milliseconds
program has been uploaded
```

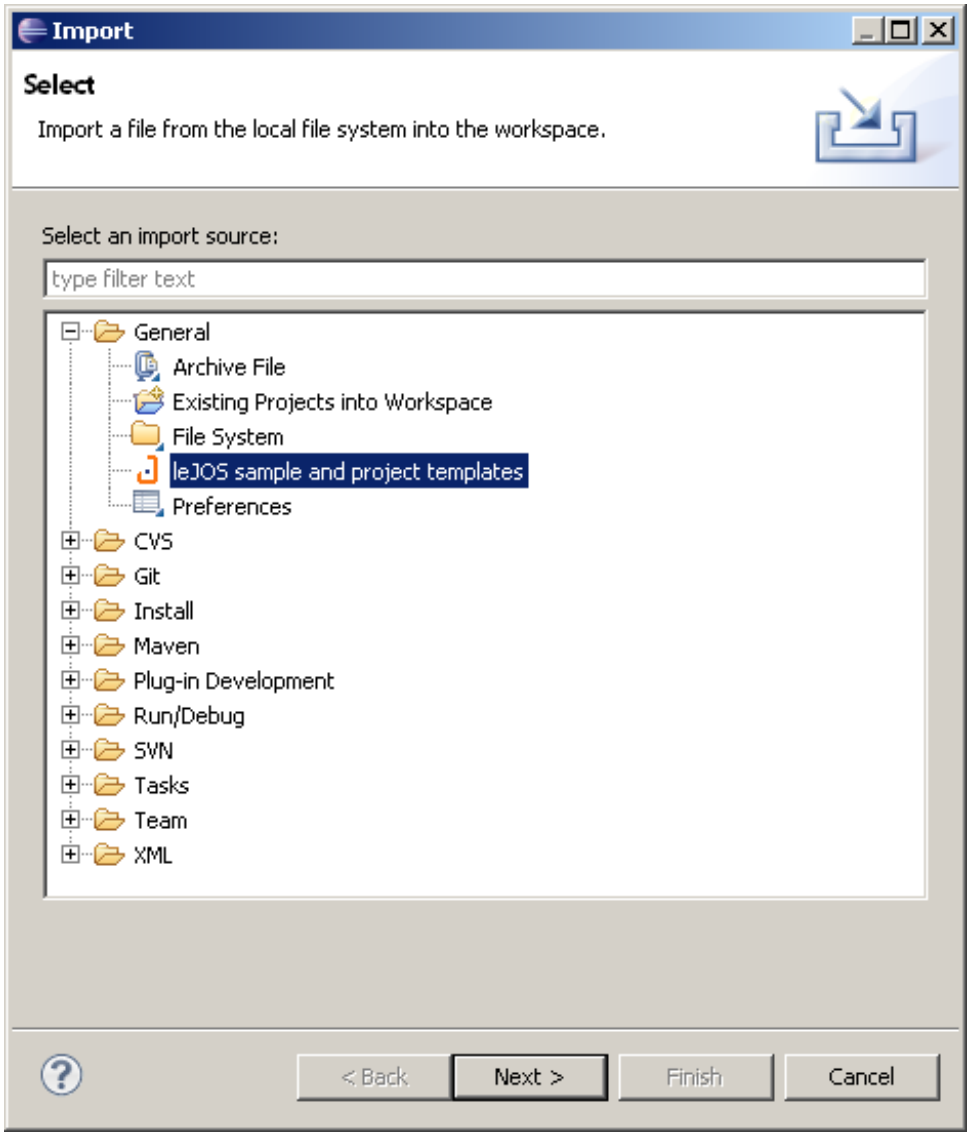
The additional information will allow you to decipher the numeric stack traces which occur on the LCD screen of the NXT when an exception happens.

Importing the samples and examples into Eclipse

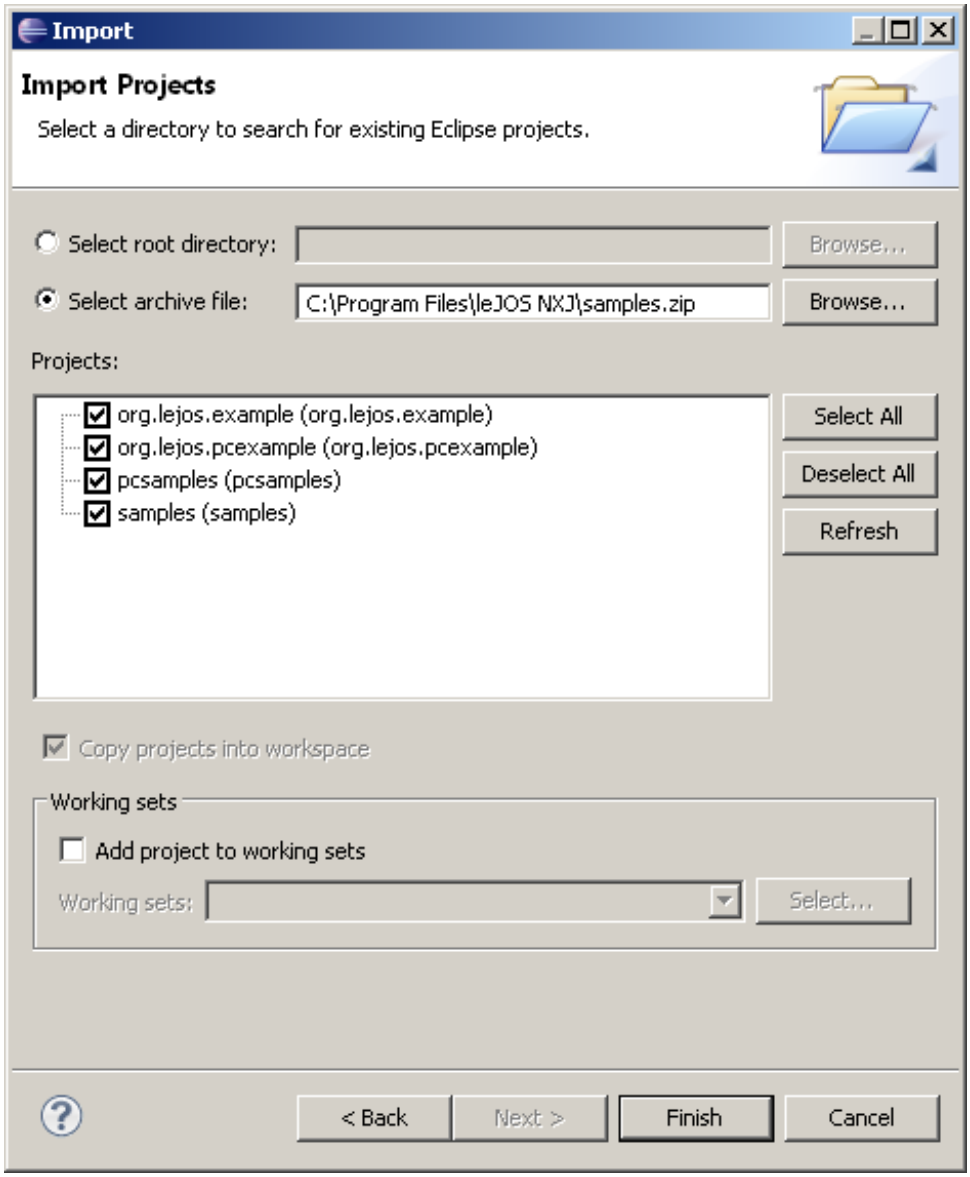
The plugin offers an import wizard to import the sample- and example-projects that come with leJOS into Eclipse. Select "File"→"Import..." in the menu.



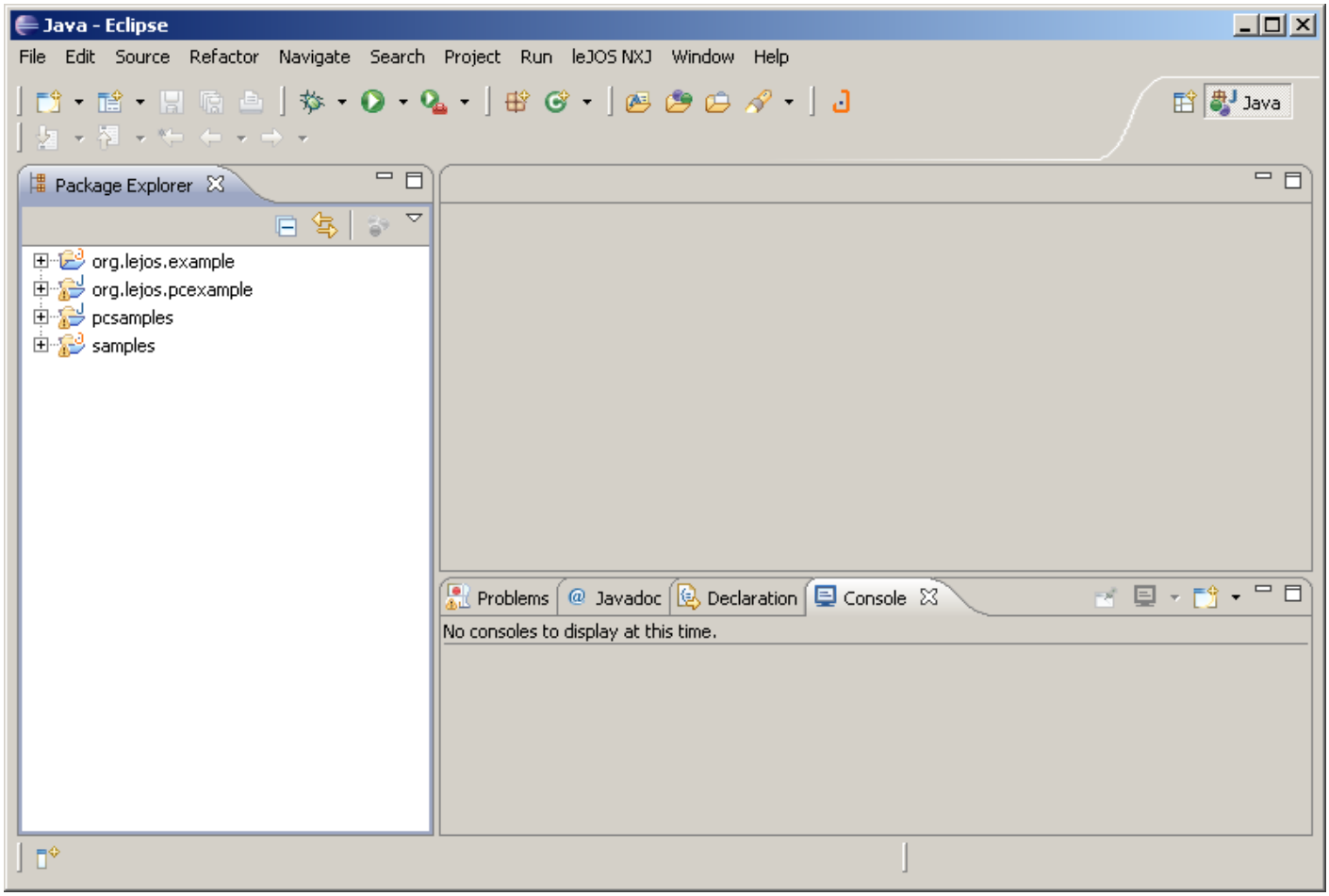
Unfold the General category and select "leJOS samples and project templates".



Click Next, and you will see the following dialog, which allows you to chose which projects to import. Note, that you cannot import a project, if a project with the same name already exists in the workspace.



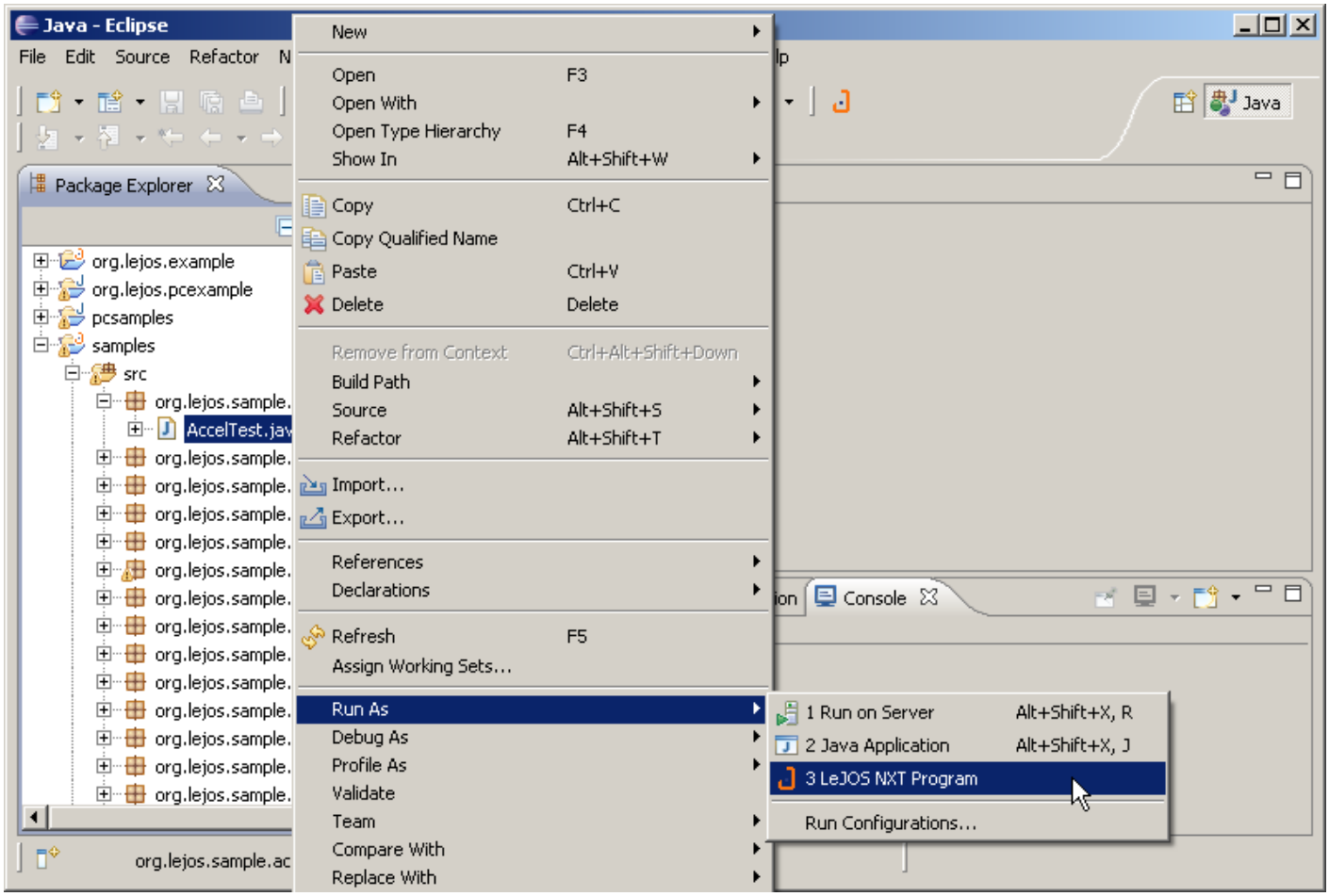
After clicking Finish, the imported projects will appear in the main window.



Back to top

Using the NXT samples project

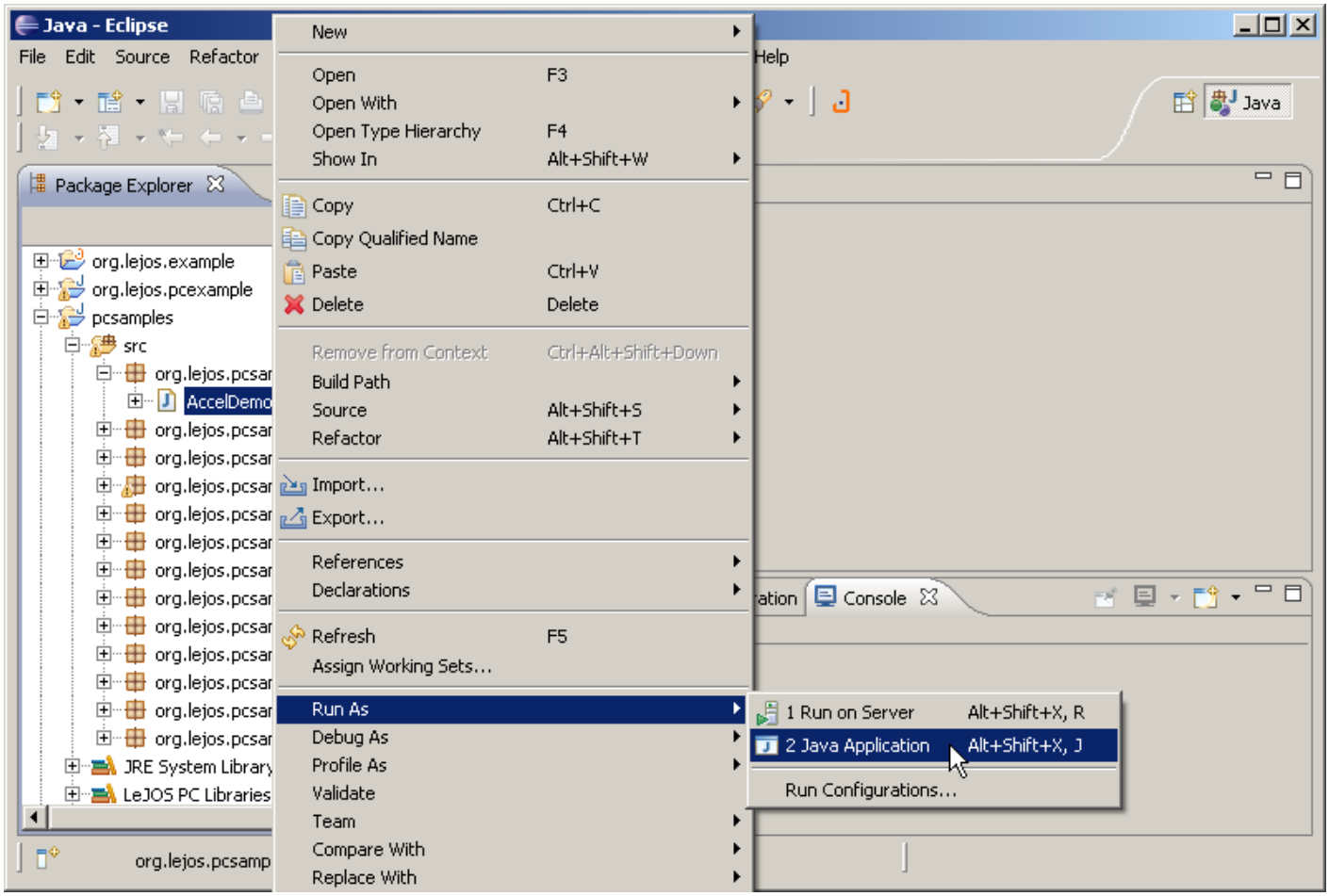
Unfold the samples project, and right-click the sample you want to start. You must select class which contains the `main` method. If you don't know, right-click the package instead. Select "Run As"→"LeJOS NXT Program" and the sample will be linked and uploaded to the NXT brick.



Back to top

Using the PC samples project

Unfold the pcsamples project, and right-click the sample you want to start. You must select class which contains the `main` method. If you don't know, right-click the package instead. Select "Run As"→"Java Application" and the sample will be started.



Back to top

Using the example projects

The example projects are a good example of how to create an Eclipse project for leJOS that also includes an Ant build file. Beside the build.xml, the projects are identical to the ones created by the project wizards described above.

org.lejos.example is an example of a project for NXT-side programs and hence the build.xml includes targets for linking and uploading the program. org.lejos.pcexample is an example of a project for PC-side programs that remote control the NXT. The Ant script automatically includes all JAR files necessary to do that in the project's classpath.

To use them as a template, import them as described above and rename the project. It is good practice to include the name of your organization in the project name to avoid name clashes with other projects in the work space. However if you are the only user of a project, this is not necessary. Then edit the build.properties file, which contains all important parameters, e.g. the name of the main class.

In order to try the Ant script from within Eclipse, right-click on build.xml and select "RunAs"→"Ant Build". In case of the example project, you should see the verbose output from the linker in the Eclipse console window and the binary will be uploaded to the NXT and run.

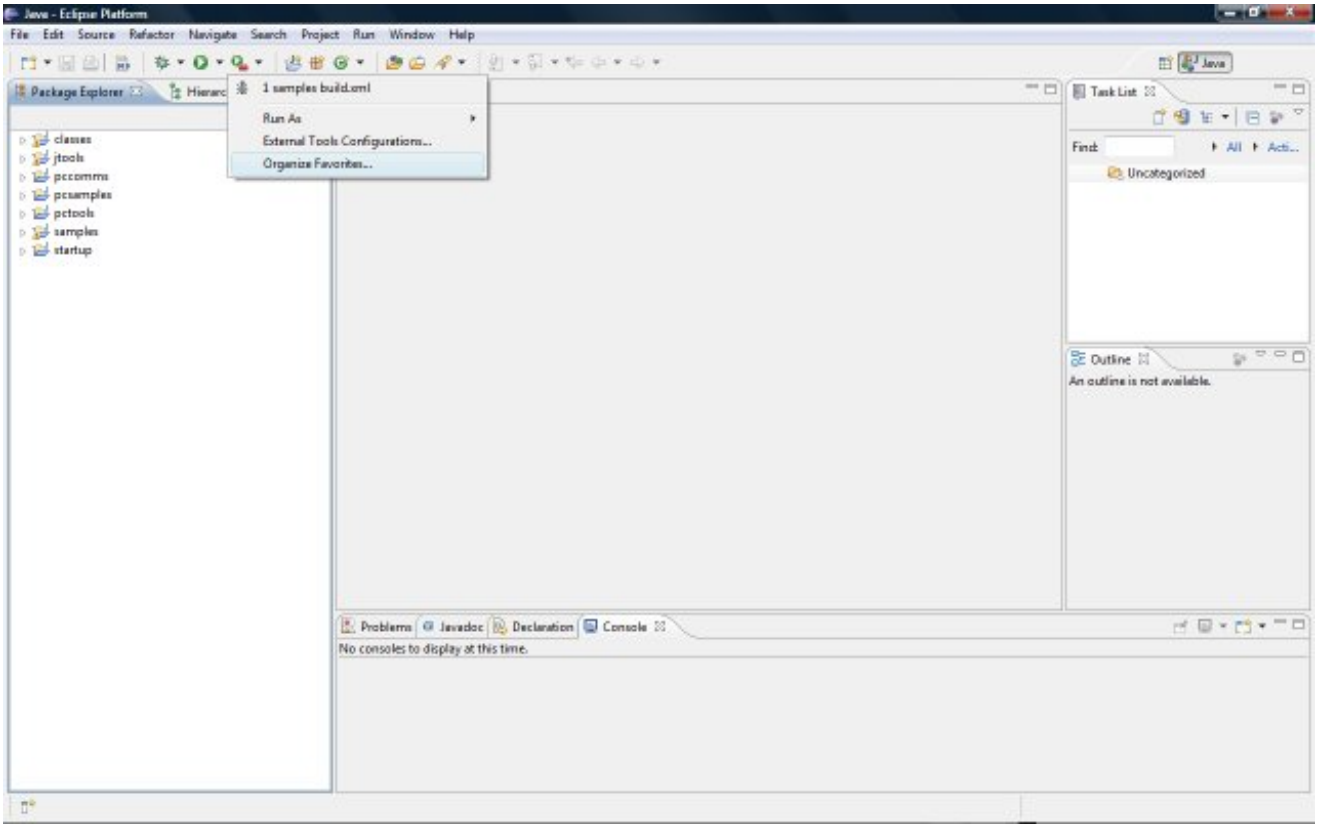
If you want to run a different ant target, right-click on build.xml and select "Run As"→"Ant Build...".

Back to top

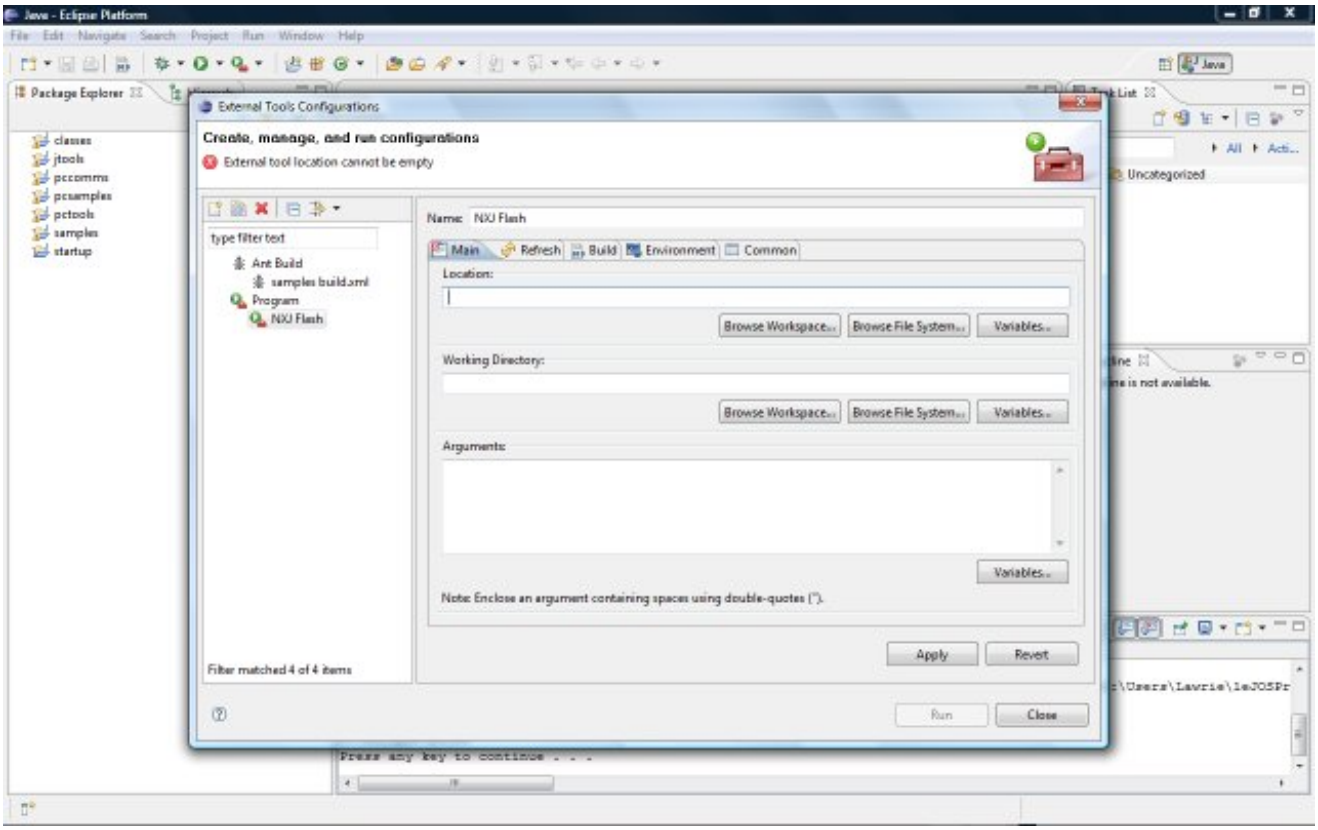
Setting up the leJOS GUI Tools

leJOS NXJ comes with a set of GUI tools that can be using for flashing the firmware, exploring files on the NXT, monitoring and debugging programs on the NXT, downloading data logs, etc. It is useful to be able to run these tools within Eclipse, and to do this you need to set them up as external tools.

To create external tools click on the down array next to the Run External Tools Icon on the toolbar and select "External Tools Configuration":



Then select Program and click on the New Launch Configuration button and you should see:



Type the name of your configuration, such as NXJ Flash and click on Browse File System and find the bin directory of where you installed leJOS NXJ and select nxjflashg.bat.

If you have multiple installations of leJOS you can go to the Environment tab and set NXJ_HOME to the one you wish to use.

Other GUI tools you should set up include:

- nxjbrowse - an Explorer for NXJ files
- nxjconsoleviewer - GUI viewer for RConsole debug output
- nxjmonitor - Remote monitoring of programs running on the NXT
- nxjdataviewer - GUI tool to download data logs from NXT
- nxjcontrol - a GUI tool that combines the function of all of the above tool, and adds a few more functions.

You may also want to set up command line tools such as:

- nxjflash - command line firmware flash tool
- nxjconsole - command line viewer for RConsole debugging output
- nxjsocketproxy - proxy for communicating with PC and Internet program using sockets