# BlueJ

## Useful Configuration Options and Lab Setup

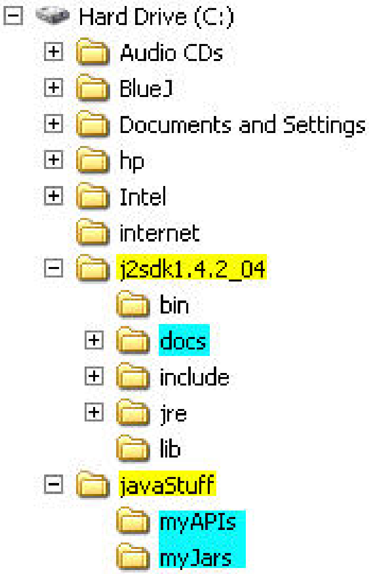
**last modified 07.24.04**

**(this page as a pdf file)**

Ok, do these in order. These steps will enable you to get all files you'll need to write programs in Java and use BlueJ to do so.

**1.** Download and install Sun's Java 2 (1.4.2 - choose the SDK, not the JRE)
**2.** Download and install BlueJ
**3.** Create a folder on your hard drive that you won't move (I'll use `c:\javaStuff` for all of my examples).
**4. Unzip** this zip file of jars right into the `c:\javaStuff` folder. You'll now have a `c:\javaStuff\myJars` folder with all the jars in it. What's in the jars zip? What is zip?
**5. Unzip** this zip file of api's right into the `c:\javaStuff` folder. You'll now have a `c:\javaStuff\myAPIs` folder with all the APIs in it. What's in the api's zip?
**6.** Optional, but highly recommended (otherwise BlueJ will go to the internet each time you need the APIs) - go to http://java.sun.com/docs/ and get Sun's API
to put on your local hard drive. **Unzip** it into the folder that was created on your hard drive as a result of doing step 1(it's called something like `c:\j2sdk1.4.2_04`). It'll create a `docs` folder.

Here is what your hard drive now looks like:

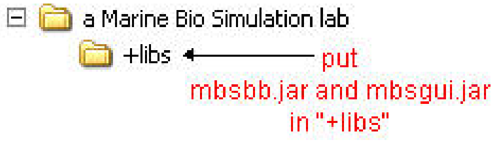


```
⊟ Hard Drive (C:)
    ⊞ Audio CDs
    ⊞ BlueJ
    ⊞ Documents and Settings
    ⊞ hp
    ⊞ Intel
      internet
    ⊟ j2sdk1.4.2_04
          bin
        ⊞ docs
        ⊞ include
        ⊞ jre
          lib
    ⊟ javaStuff
          myAPIs
          myJars
```

---

Ok, physically, your computer has what it needs.
BlueJ will now run. In order to use your own jar files you'll have to enable BlueJ to see them.
There are different ways to do that; I will only give you what I believe is the easist way.
BlueJ works on a folder system; everything that is part of one project must be in a single folder (that's a good thing).
If you have any jars that the project needs, just create a "+libs" folder within the project folder and put any jars you need in it.

So, for an MBS project, it would look like this:



```
⊟ a Marine Bio Simulation lab
      +libs
```

### Starting your students on a new project each time

If you just create a project with all files needed for it and then place it on a network drive, your students could just copy it into their account and begin. Here are 3 projects with the generic setup that I use:

Karel Folder with which to begin a project

MBS Folder with which to begin a project

MBS Folder with which to begin a project (dynamic population)

### Opening BlueJ Projects

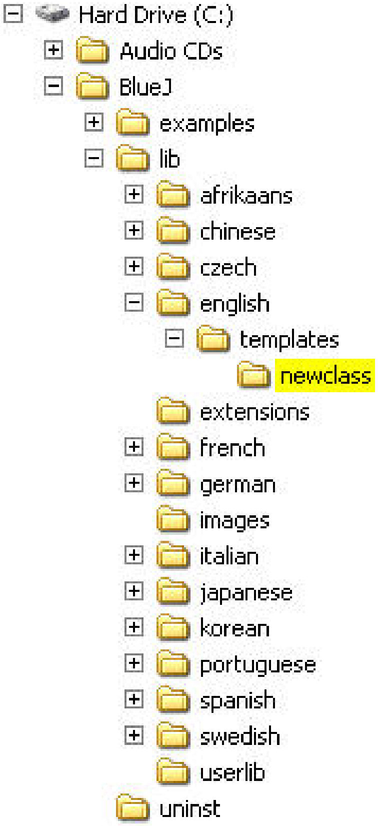
1) project → open project...

### Opening Non-BlueJ Projects

1) project → open non-BlueJ...
2) click on folder that contains your project files (don't go into the folder, just select it)
3) click on "open in BlueJ"
note: from now on, it's a BlueJ project (you'll see some other files in your folder now - they're for BlueJ).
When you go to open this project the next time, just do a normal "open project..."

### Editing the .tmpl files

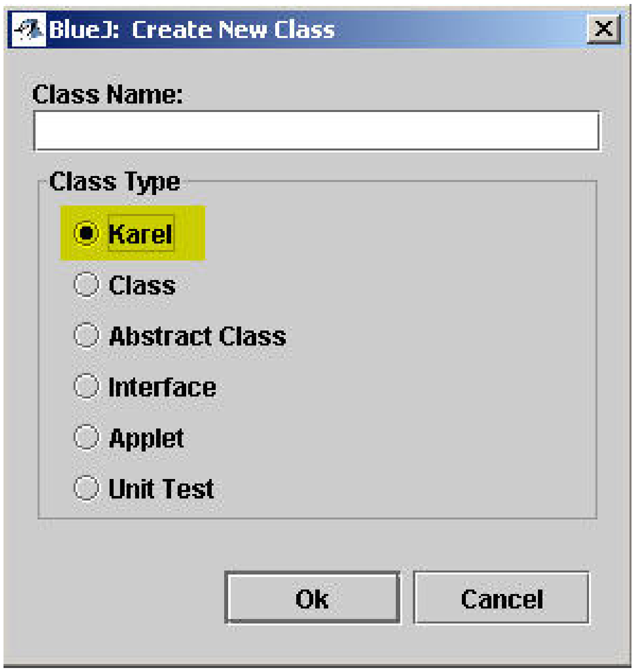
The .tmpl files control what a default class, interface, abstract class, etc. will look like when you choose "New Class". Just go to the "newclass" folder within the BlueJ install folder. Here is where it's at:



```
⊟ Hard Drive (C:)
    ⊞ Audio CDs
    ⊟ BlueJ
        ⊞ examples
        ⊟ lib
            ⊞ afrikaans
            ⊞ chinese
            ⊞ czech
            ⊟ english
                ⊟ templates
                      newclass
              extensions
            ⊞ french
            ⊞ german
              images
            ⊞ italian
            ⊞ japanese
            ⊞ korean
            ⊞ portuguese
            ⊞ spanish
            ⊞ swedish
              userlib
          uninst
```

You'll see some .tmpl files. They are just text files. You can open them and edit as you wish. Here are 2 that you can use that I just add to the newclass folder:

Karel.tmpl and newclass.tmpl If you add these, you'll see this when starting a new class within the environment:



Class Name:

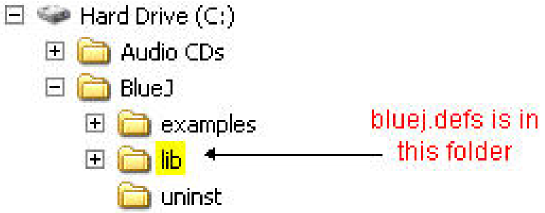
Class Type

● Karel
○ Class
○ Abstract Class
○ Interface
○ Applet
○ Unit Test

## Editing the bluej.defs file

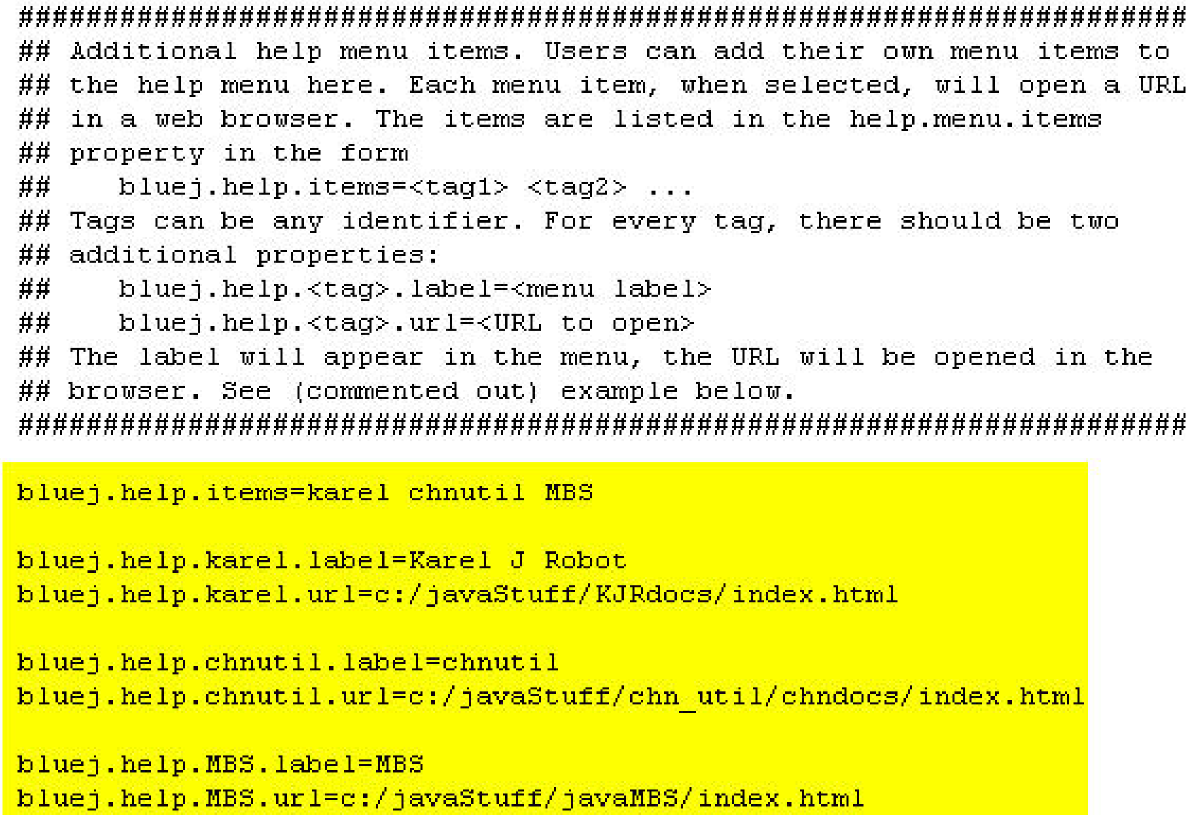
The `bluej.defs` file is a text file. It controls the default settings for many things in BlueJ. Open it, read the comments, then edit what you will. I edit the section that controls what options are listed under the "Help" menu. My **`bluej.defs`** file (you can replace the installed one with it - if you kept to my exact folder structure above, you won't need to edit this version I created - see pic below
for exactly what I've edited in the file). If you placed the APIs where I suggested above and you replaced the `bluej.defs` file with
mine, you'll be able to open the APIs directly from within BlueJ(see the BlueJ Help menu graphic below)

Here is where the `bluej.defs` file is located:



bluej.defs is in this folder

Here is the part of the `bluej.defs` file that I edited:

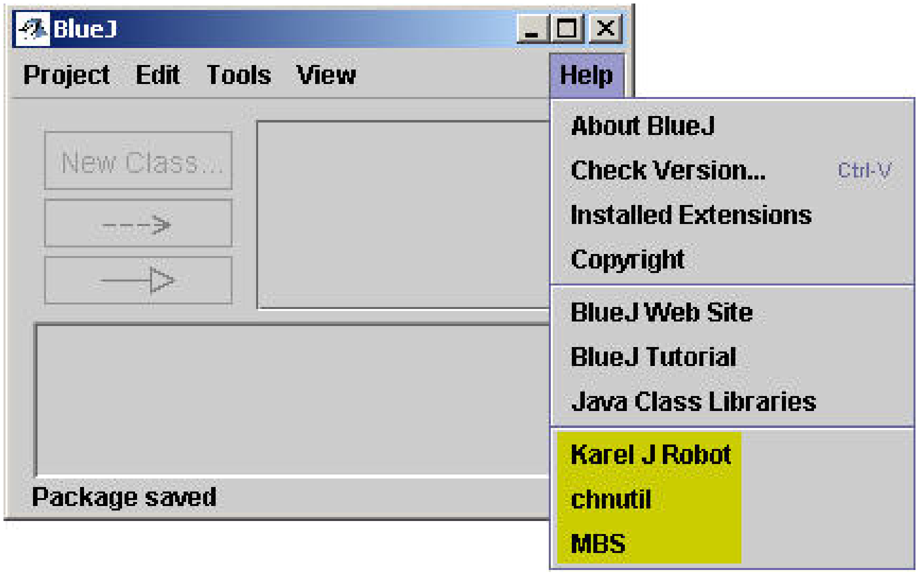


```
#######################################################################
## Additional help menu items. Users can add their own menu items to
## the help menu here. Each menu item, when selected, will open a URL
## in a web browser. The items are listed in the help.menu.items
## property in the form
##    bluej.help.items=<tag1> <tag2> ...
## Tags can be any identifier. For every tag, there should be two
## additional properties:
##    bluej.help.<tag>.label=<menu label>
##    bluej.help.<tag>.url=<URL to open>
## The label will appear in the menu, the URL will be opened in the
## browser. See (commented out) example below.
#######################################################################

bluej.help.items=karel chnutil MBS

bluej.help.karel.label=Karel J Robot
bluej.help.karel.url=c:/javaStuff/KJRdocs/index.html

bluej.help.chnutil.label=chnutil
bluej.help.chnutil.url=c:/javaStuff/chn_util/chndocs/index.html

bluej.help.MBS.label=MBS
bluej.help.MBS.url=c:/javaStuff/javaMBS/index.html
```

#Here is the code so you can COPY it (thanks, Mike!)

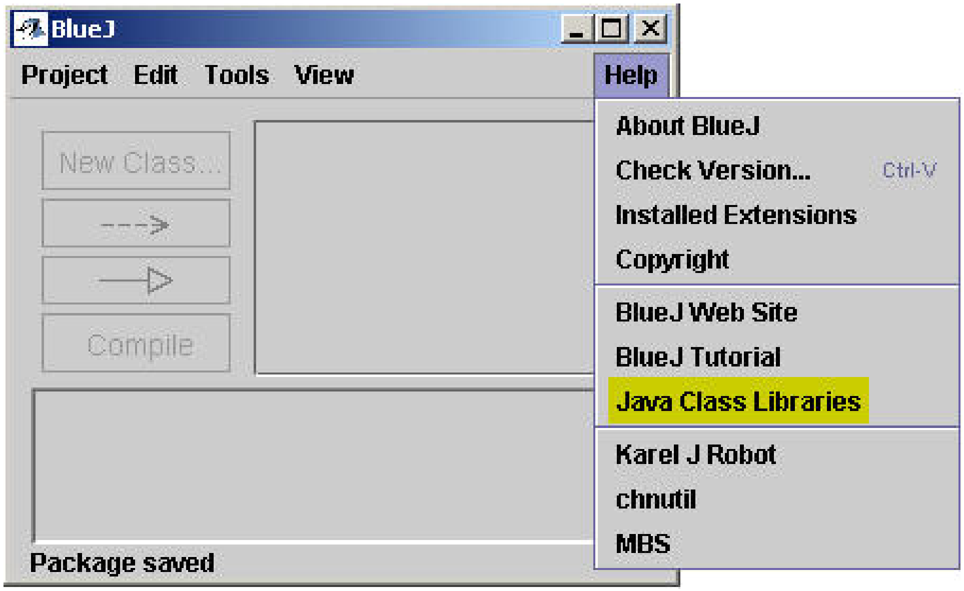
```
bluej.help.items=karel chnutil MBS

bluej.help.karel.label=Karel
bluej.help.karel.url=c:/javaStuff/KJRdocs/index.html

bluej.help.chnutil.label=chnutil
bluej.help.chnutil.url=c:/javaStuff/chn_util/chndocs/index.html

bluej.help.MBS.label=MBS
bluej.help.MBS.url=c:/javaStuff/javaMBS/index.html
```

If you edit the file as above, you'll see this show up in BlueJ:



### Setting BlueJ to use a local copy of Sun's API instead of going out to the internet

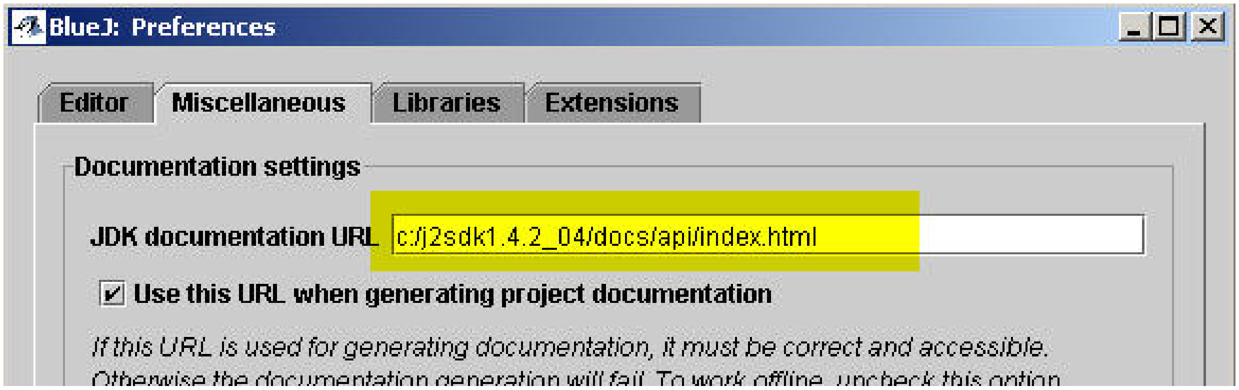
By default, BlueJ will go out to the internet each time when you select "Java Class Libraries" from the Help menu:



This is fine if you have a fast connection and your lab is always connected (don't I wish). Change it so that BlueJ uses the local copy of the API (if you took the suggestion and did step 6 at the top of this page). Just go to

`Tools` → `Preferences...` → `Miscellaneous`

and change the URL to the location of the index.html file for the API.



Documentation settings

JDK documentation URL c:/j2sdk1.4.2_04/docs/api/index.html

☑ Use this URL when generating project documentation

*If this URL is used for generating documentation, it must be correct and accessible. Otherwise the documentation generation will fail. To work offline, uncheck this option*

### What's in the jars zip?

mbsbb.jar, mbsgui.jar, KarelTheRobot.jar, apcslib.jar, chnutil.jar, objectdraw.jar, EasyReader.class, EasyWriter.class

### What's in the api's zip?

The APIs for MBS, Karel, chnutil, EasyReader, EasyWriter, Owen Astrachan's AP Java subset

### Objectdraw with BlueJ?

(see http://applecore.cs.williams.edu/~cs134/eof/library/readmes/objectdraw4BlueJ.pdf for details)

### What is zip?

You can read about zipping/unzipping and get a free evaluation of WinZip