Developing FISH on Windows

RKH 23/12/08, 16/03/17

SOFTWARE TO INSTALL ON YOUR PC

Install the following; swigwin and eclipse do not appear in "add/remove programmes" so could, I think be copied over from another machine. The jfreechart graphics library comes in from a library on the repository set up by Jonathan.

Tortoise SVN – subversion manager, downloads files from central repository and tracks changes so you can manage uploads back to the repository, http://tortoisesvn.net/downloads.

Eclipse - software development platform for Java etc, http://www.eclipse.org/downloads/ Eclipse Classic, with the purple icon  . I put it in C:\FISH4\ECLIPSE\.

Subclipse – provides support for subversion within Eclipse. Start Eclipse then: help, software updates, available software, add site, http://subclipse.tigris.org/update_1.4.x, then select box and install. See www.subclipse.tigris.org, but note that latest Eclipse screens are not quite as in the example given.

Photran – not absolutely necessary, but lets you edit Fortran inside Eclipse, seems to have some useful global editing capabilities, same method as Subclipse above, enter url http://www.eclipse.org/photran/download.php.

https://eclipse.org/photran/download.php

Photran 9.1 requires Eclipse 4.5 (Mars) and CDT 8.7.

- If you have not installed Photran before, or if you run into any problems during installation, please read the installation instructions.
- If you are familiar with the installation procedure, Photran can be installed from the Eclipse Mars update site:
  http://download.eclipse.org/releases/mars

Swig – connects scripting languages with C, http://www.swig.org/download.html, I installed in C:\FISH4\SWIGWIN\

GrWin – pgplot drivers for windows, http://spdg1.sci.shizuoka.ac.jp/grwinlib/english/, install in C:\GrWin. [Hmm I also have stuff in C:\FISH4\PGPLOT\.... but not sure where this came from or whether it is needed here!]

Java - http://java.sun.com/javase/downloads/index.jsp , I got the "Java SE Development Kit (JDK) 6 Update 11" which seems to do the job.

MinGW – C and Fortan compiler, http://www.mingw.org/old/download.shtml, there are a lot of downloads to choose from, I used "Automated MinGW installer", then chose MinGW-5.12.4.exe,

download & run, asked for G++ and java compilers [not sure if the latter was needed], which it then downloaded, unzipped and copied into C:\MinGW\.

SETTING UP THE SOFTWARE

Start Eclipse, if it asks for a "workspace" use, e.g." C:\fish4\eclipse_workspace2".

First time, do I think, [it's hard to start over again from scratch to check], "new workspace", "new project wizard", find SVN in explorer tree, create new repository, enter https://svn.diamond.ac.uk/subversion/ncd/sas-software/fish and enter your federal id & password when prompted, select FISH folder, highlight TRUNK, checkout as a "project in the workspace", then rename TRUNK to for example FISH4a.

Jonathan has 5 files in the download, in three different directories, which control the project locations and its "makefile", which all have extensions ".sample". Make local copies of these and remove the extension. This is easier said than done as Windows Explorer does not let you rename a filename with a leading dot, so some messing about in a cmd window is required!

The files are these, all in C:\FISH4\eclipse_workspace2\Fish4a\

.classpath
.project
src\makefile.win
.externalToolBuilders\make clean.launch
.externalToolBuilders\make.launch

Annotated copies of the working Windows versions of these are below.

Various file paths and directory names have to be edited, particularly in makefile.win. In places where Windows "copy" is used the path must use \ not /.

Initial debugging may be simpler if you "clean" the project in Eclipse, and then run the makefile from a cmd window in directory FISH4a\src\ using:

C:\MinGw\bin\MinGW32-make.exe –f makefile.win

It all looks very simple when it is working and the project is built from inside Eclipse, but it took a while to debug the process! For some strange reason a normal shortcut and Eclipse expect the file FISH.DLL in different places. Initially when running the project from inside Eclipse an earlier version of the Fortran part appeared! To make verifiable changes of no consequence to the code, edit the version string in FISH.JAVA or the "welcome" message in FISH1.F. Note put "-console" after the shortcut to the executable to see the fortran output, i.e. run C:\fish4\eclipse_workspace2\FISH4a\fish.exe –console.

MinGW32-make parameters (double dash are underscore)

Detail manual: https://www.gnu.org/software/make/manual/make.pdf

```
 -b, -m                      Ignored for compatibility.
 -B, --always-make           Unconditionally make all targets.
 -C DIRECTORY, --directory=DIRECTORY
                             Change to DIRECTORY before doing anything.
 -d                          Print lots of debugging information.
 --debug[=FLAGS]             Print various types of debugging information.
 -e, --environment-overrides
                             Environment variables override makefiles.
 --eval=STRING               Evaluate STRING as a makefile statement.
 -f FILE, --file=FILE, --makefile=FILE  Read FILE as a makefile.
 -h, --help                  Print this message and exit.
 -i, --ignore-errors         Ignore errors from recipes.
 -I DIRECTORY, --include-dir=DIRECTORY
                             Search DIRECTORY for included makefiles.
 -j [N], --jobs[=N]          Allow N jobs at once; infinite jobs with no arg.
 -k, --keep-going            Keep going when some targets can't be made.
 -l [N], --load-average[=N], --max-load[=N]
                             Don't start multiple jobs unless load is below N.
 -L, --check-symlink-times   Use the latest mtime between symlinks and target.
 -n, --just-print, --dry-run, --recon
                             Don't actually run any recipe; just print them.
 -o FILE, --old-file=FILE, --assume-old=FILE
                             Consider FILE to be very old and don't remake it.
 -p, --print-data-base       Print make's internal database.
 -q, --question              Run no recipe; exit status says if up to date.
 -r, --no-builtin-rules      Disable the built-in implicit rules.
 -R, --no-builtin-variables  Disable the built-in variable settings.
 -s, --silent, --quiet       Don't echo recipes.
 -S, --no-keep-going, --stop
                             Turns off -k.
 -t, --touch                 Touch targets instead of remaking them.
 --trace                     Print tracing information.
 -v, --version               Print the version number of make and exit.
 -w, --print-directory       Print the current directory.
 --no-print-directory        Turn off -w, even if it was turned on implicitly.
 -W FILE, --what-if=FILE, --new-file=FILE, --assume-new=FILE
                             Consider FILE to be infinitely new.
 --warn-undefined-variables  Warn when an undefined variable is referenced.
```

Files to be edited

.classpath

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
	<classpathentry kind="src" path="java">
		<attributes>
			<attribute
name="org.eclipse.jdt.launching.CLASSPATH_ATTR_LIBRARY_PATH_ENTRY"
value="FISH/bin"/>
		</attributes>
	</classpathentry>
	<classpathentry kind="con"
path="org.eclipse.jdt.launching.JRE_CONTAINER"/>
	<classpathentry kind="lib"
path="C:/fish4/eclipse_workspace2/FISH4a/lib/jcommon-1.0.5.jar">
		<attributes>
			<attribute name="javadoc_location"
value="http://www.jfree.org/jcommon/api/"/>
		</attributes>
	</classpathentry>
	<classpathentry kind="lib"
path="C:/fish4/eclipse_workspace2/FISH4a/lib/jfreechart-1.0.2.jar">
		<attributes>
			<attribute name="javadoc_location"
value="http://www.jfree.org/jfreechart/api/javadoc/"/>
		</attributes>
	</classpathentry>
	<classpathentry kind="output" path="bin"/>
</classpath>
```

.project

```
<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
	<name>FISH4a</name>
	<comment></comment>
	<projects>
	</projects>
	<buildSpec>
		<buildCommand>

	<name>org.eclipse.ui.externaltools.ExternalToolBuilder</name>
			<triggers>clean,</triggers>
			<arguments>
				<dictionary>
					<key>LaunchConfigHandle</key>

	<value>&lt;project&gt;/.externalToolBuilders/make
clean.launch</value>
				</dictionary>
			</arguments>
		</buildCommand>
		<buildCommand>

	<name>org.eclipse.ui.externaltools.ExternalToolBuilder</name>
			<triggers>auto,full,incremental,</triggers>
			<arguments>
				<dictionary>
					<key>LaunchConfigHandle</key>
```

```
      <value>&lt;project&gt;/.externalToolBuilders/make.launch</value>
                        </dictionary>
                  </arguments>
            </buildCommand>
            <buildCommand>
                  <name>org.eclipse.jdt.core.javabuilder</name>
                  <arguments>
                  </arguments>
            </buildCommand>
      </buildSpec>
      <natures>
            <nature>org.eclipse.jem.workbench.JavaEMFNature</nature>
            <nature>org.eclipse.jdt.core.javanature</nature>
            <nature>org.eclipse.jem.beaninfo.BeanInfoNature</nature>
      </natures>
</projectDescription>
```

src\makefile.win [Make sure all paths and Java version numbers are correct ! Tab indented lines only run if files in line above are out of date, or a full build is specified.]

```
fish.dll: fish.o fish_wrap.o FISH1.o FISH2.o FISH3.o FISH4.o FISH5.o
FISH6_DUMMY.o FISH7.o FISHDIM.PAR FISHGRAPH.o FISHMODEL.o fish_getwir.o
fishqp1.o fishqp2.o fishqp3.o fishqpzero.o fishsmear.o fishsysG77.o
launcher.exe fish.png
      c:/MinGW/bin/g77.exe -D_JNI_IMPLEMENTATION_ -Wl,--kill-at,--
subsystem,windows -shared *.o -lg2c -o fish.dll -mwindows -I.
      copy fish.dll ..\bin\fish.dll   [NOTE \ not / for windows copy. This dll for shortcut]
      copy fish.dll ..\fish.dll       [this dll  is for Eclipse]
      mkdir ..\bin\fish
      copy fish.png ..\bin\fish\fish.png   [copies the icon!]

fish.o fish_wrap.o: fish.c fish_wrap.c fish.h
      c:/MinGW/bin/gcc.exe -D_JNI_IMPLEMENTATION_ -c -Ic:/Program\
Files/Java/jdk1.6.0_11/include -Ic:/Program\
Files/Java/jdk1.6.0_11/include/win32 fish.c fish_wrap.c

FISH1.o FISH2.o FISH3.o FISH4.o FISH5.o FISH6_DUMMY.o FISH7.o FISHGRAPH.o
FISHMODEL.o fish_getwir.o fishqp1.o fishqp2.o fishqp3.o fishqpzero.o
fishsmear.o fishsysG77.o: FISH1.F FISH2.F FISH3.F FISH4.F FISH5.F
FISH6_DUMMY.F FISH7.F FISHDIM.PAR FISHGRAPH.F FISHMODEL.F PGCONTROL.INC
PGPLOT.INC fhelpfit.txt fish_getwir.f fishqp1.f fishqp2.f fishqp3.f
fishqpzero.f fishsmear.f fishsysG77.f rgb.txt
      c:/MinGW/bin/g77.exe -c -s -fno-automatic -finit-local-zero -fno-
backslash -ffixed-line-length-80 *.f *.F -mwindows -I.

fish_wrap.c: fish.h fish.i
      C:\fish4\swigwin\swig -outdir ../java/fish -package fish -module Core
-java fish.i

clean:
      -rm fish.dll launcher.exe *.o fish_wrap.c ../hs_*.log

launcher.exe: winlauncher.c icon.o
      c:/MinGW/bin/gcc.exe -Wl,--subsystem,windows -mwindows winlauncher.c
icon.o -o launcher.exe
      copy launcher.exe ..\fish.exe
```

```
icon.o: fish.ico fish.rc
      windres fish.rc icon.o
```

.externalToolBuilders\make clean.launch

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<launchConfiguration
type="org.eclipse.ui.externaltools.ProgramBuilderLaunchConfigurationType">
<stringAttribute key="org.eclipse.debug.core.ATTR_REFRESH_SCOPE"
value="${project}"/>
<booleanAttribute key="org.eclipse.debug.core.appendEnvironmentVariables"
value="true"/>
<booleanAttribute key="org.eclipse.debug.ui.ATTR_LAUNCH_IN_BACKGROUND"
value="false"/>
<booleanAttribute key="org.eclipse.ui.externaltools.ATTR_BUILDER_ENABLED"
value="true"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_LOCATION"
value="C:\MinGW\bin\mingw32-make.exe"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_RUN_BUILD_KINDS"
value="clean"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_TOOL_ARGUMENTS"
value="-fMakefile.win clean"/>
<booleanAttribute
key="org.eclipse.ui.externaltools.ATTR_TRIGGERS_CONFIGURED" value="true"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_WORKING_DIRECTORY"
value="${workspace_loc:/FISH4a/src}"/>
</launchConfiguration>
```

.externalToolBuilders\make.launch

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<launchConfiguration
type="org.eclipse.ui.externaltools.ProgramBuilderLaunchConfigurationType">
<booleanAttribute key="org.eclipse.debug.core.appendEnvironmentVariables"
value="true"/>
<booleanAttribute key="org.eclipse.debug.ui.ATTR_LAUNCH_IN_BACKGROUND"
value="false"/>
<booleanAttribute key="org.eclipse.ui.externaltools.ATTR_BUILDER_ENABLED"
value="true"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_LOCATION"
value="C:\MinGW\bin\mingw32-make.exe"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_RUN_BUILD_KINDS"
value="full,incremental,auto,"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_TOOL_ARGUMENTS"
value="-fMakefile.win"/>
<booleanAttribute
key="org.eclipse.ui.externaltools.ATTR_TRIGGERS_CONFIGURED" value="true"/>
<stringAttribute key="org.eclipse.ui.externaltools.ATTR_WORKING_DIRECTORY"
value="${workspace_loc:/FISH4a/src}"/>
</launchConfiguration>
```

16/3/17 on screen Makefile in Eclipse from 2008 has

```
libfish.so: fish.o fish_wrap.o FISH1.o FISH2.o FISH3.o FISH4.o FISH5.o
FISH6_DUMMY.o FISH7.o FISHDIM.PAR FISHMODEL.o fish_getwir.o fishqp1.o
fishqp2.o fishqp3.o fishqpzero.o fishsmear.o fishsysG77.o fish.png
      gcc -shared *.o -lg2c -o libfish.so
      cp libfish.so ../bin
      cp fish.png ../bin/fish

fish.o fish_wrap.o: fish.c fish_wrap.c fish.h
      gcc -c -I/usr/lib/jvm/java/include/ -
I/usr/lib/jvm/java/include/linux/ fish.c fish_wrap.c

FISH1.o FISH2.o FISH3.o FISH4.o FISH5.o FISH6_DUMMY.o FISH7.o FISHMODEL.o
fish_getwir.o fishqp1.o fishqp2.o fishqp3.o fishqpzero.o fishsmear.o
fishsysG77.o: FISH1.F FISH2.F FISH3.F FISH4.F FISH5.F FISH6_DUMMY.F FISH7.F
FISHDIM.PAR FISHMODEL.F PGCONTROL.INC PGPLOT.INC fhelpfit.txt fish_getwir.f
fishqp1.f fishqp2.f fishqp3.f fishqpzero.f fishsmear.f fishsysG77.f rgb.txt
      g77 -c -s -fno-automatic -finit-local-zero -fno-backslash -ffixed-
line-length-80 *.f *.F -I.

fish_wrap.c: fish.h fish.i
      swig -outdir ../java/fish -package fish -module Core -java fish.i

clean:
      -rm fish *.o fish_wrap.c ../hs_*.log
```

29/3/17 Java8 seems to have introduced a problem :

Building from 32bit Eclipse Ganymede, either FISH4a or FISH5a, gives error

“The project was not built since its build path is incomplete. Cannot find the class file for java.io.ObjectInputStream. Fix the build path then try building this project”

“The type java.io.ObjectInputStream cannot be resolved. It is indirectly referenced from required .class files”

Lots on this in Google, e.g.

---

https://bugs.openjdk.java.net/browse/JDK-8155223 2016-08-03 04:11 pasted comments from other bug reports :

Old versions of Tomcat 6 JSP compiler don't seem to be aware of JDK 8 constant pool enhancements - eg. method handles. New code in JDK 8u is using a method handle instead of creating an anonymous class. This will cause the method handle to be listed in the constant pool and the eclipse compiler will choke on this - https://bz.apache.org/bugzilla/show_bug.cgi?id=56613

End users would have had the same problems for any version of JDK 8 if their JSPs referenced any classes containing method handles. The failure you can observe here is unearthed only because a JSP

is accessing ObjectInputStream which happens to contain a MethodHandle now.

I am not certain this is JDK problem - this version of tomcat really should not be used together with JDK 8. One could run into this kind of problems by referencing any other JDK class with JDK 8 only constant pool enhancements. Upgrading to the latest Tomcat 6.x version(or later family versions) might solve issue. In any case, Tomcat 6.x versions are almost end of life. Best to upgrade to a newer version of Tomcat : http://tomcat.apache.org/tomcat-60-eol.html

---