# jCompile User Guide

First version: 22 August 2003
Last updated for version 1.02 on 19 October 2003

## 1.0 What is jCompile?

Welcome to jCompile - the Personal Java compiler for the P800 smartphone and Psion netBook / Series 7 / Series 5mx computers. This package is not intended for novice users. For example, the P800 user interface is intended to hide the complexities of the file system from the user. We will do the opposite, using Personal Java source and compiled code extensively on the P800's file system. We assume therefore that you are an experienced user.

Note that jCompile only compiles Personal Java, which approximates to Java 1.1.8. It does not compile MIDP Java or any other Java variant. From now on we take "Java" to mean "Personal Java".

We will not teach you how to program in Java. We assume that you are already familiar with Java on other platforms. We will show you how to write, edit and compile Java code directly on the P800 or Psion. However you use this software entirely at your own risk, we will not be responsible for any problems or errors caused.

We acknowledge the trademarks and rights of Sun Microsystems and Symbian Ltd. A special word of thanks to Frank Bodmann, who wrote Javac for the Psion Series 5 on which this program is based.

## 1.1 Why compile Java code on your P800 or Psion?

Why compile Java code on your P800 smartphone or Psion computer? Because you can :-))

If you wish, you can write and deploy Java programs on your P800 or Psion without ever touching Symbian's Software Development Kits (SDKs). Clearly for the P800 you will not want to write long programs without the benefit of a keyboard. But it is perfectly feasible to write short programs from scratch or to debug longer programs. Alternatively, you could write the Java code on a PC, then transfer it to your P800.

## 1.2 What you need to compile Java programs on your P800 or Psion

The following programs are all provided as SIS files within the distribution package. As an experienced user, we assume that you understand how to install SIS files onto your smartphone or computer.

- *File Manager* by Symbian Ltd – to move files around on your P800.
- *Redirect* by Symbian Ltd – enables the standard Java output stream.
- *jText* by FreEPOC – a text editor (written in Java).
- *jCompile* by FreEPOC – the Java compiler.
- *jRun* by FreEPOC – run your Java program from the P800 Application menu or Psion

Extras bar.

You will also need to extract some Java classes from Sun's Java SDK, as explained in the Readme.txt file in the distribution zip.

## 1.3 Enabling the Console (Psion only)

To see the output from jCompile, you must enable the Console. To do this, create a file of any type called "console" and place it in the directory c:\System\Java\. You will then be able to see the output from jCompile in the file \Logs\Java\Java.txt on either the c: or d: drive where you installed jCompile.

## 2.0 Compiling the "Hello World" program.

When you installed *jCompile*, a directory c:\classes was created. In this directory you will find a simple source file *HelloWorld.java*. We will demonstrate the Java compiler using this source file.

**Step 1.** Firstly use the Task List option in *File Manager* to close all non–essential programs to free up memory (P800 only).
**Step 2.** Now start the *Redirect* program, so that we can see the outputs from the Java compiler (P800 only)
**Step 3.** Finally run *jCompile*, select the New option and navigate to the *HelloWorld.java* source file in the c:\classes directory. [P800 users note: in the file dialog do not select Browse. Owing to a bug in Symbian's Java implementation, this will cause jCompile to crash.]

The program should compile successfully and you can check this by going back to *Redirect* if you are using the P800. You should find a new file called *HelloWorld.class* in the c:\classes directory. Psion users can see the results of the compilation in \Logs\Java\Java.txt as described above.

## 2.1 Running the "Hello World" program.

Click on *HelloWorld.class* in *File Manager*. You should see the familiar "Starting" message appear briefly. Now go to *Redirect*. You should see that the program has printed "Hello World" to the standard output stream. Psion users should look at \Logs\Java\Java.txt. As you have seen, the Symbian OS recognises class files as valid Java programs and clicking in *File Manager* will launch them. We will see how to launch a Java program from the P800 Application Menu or Psion Extras bar a little later.

## 3.0 Compiling & running the "Converter" program

You should see a program *Converter.java*, also located in directory c:\classes. This program is part of Sun's Java 1.1 standard documentation. It is a generic Java program and is not optimised for the P800 or Psion. It is useful because it demonstrates a couple of quirks of Symbian's Java implementation.

Compile *Converter.java* following the same steps as for the HelloWorld program. You should find a number of additional class files in c:\classes. In *File Manager*, find the file *Converter.class* and click on it. After a brief pause, the program should launch.

As you can see, the Converter program converts between imperial and metric distances. Enter a distance and select a unit of measurement. You should see the results of the conversion appear in the other half of the screen. Since no Menu is available in this program (remember, it was not written specifically for the P800) there is no obvious way to terminate the program.

**Note for P800 users:** You will need to go to *File Manager* and select the Task List. Highlight the Converter program and press Close File. Java class files that you have launched from *File Manager* do not appear as running programs in some P800 utility programs such as *Switcher*. However they will always appear in *File Manager's* Task List and can be terminated from there.

**Note for Psion users:** Symbian have provided a Java Sweep utility to terminate Java programs that cannot be closed from a user-selectable option. Copy the program *sweeperPsion.exe* to any folder on your Psion. To terminate *Converter.class*, highlight *sweeperPsion.exe* and press Enter.

## 3.1 Using the jRun program

It can be inconvenient to launch Java class files from the P800 or Psion File Manager. Ideally, these programs should launch from the P800 Application list or Psion Extras bar. We have therefore provided a very simple Java wrapper which you can use. Find *jRun* in the Applications list or Extras bar and click on it. It should launch the same Converter program we used above.

In fact, the *jRun* program will launch whichever Java class file is shown in the *jRun.txt* file. This file is located in the directory \system\apps\jRun\ on the disk where you installed the *jRun* program. Find this file in *File Manager* and click on it to see the contents. It will display

```
-cp c:\classes Converter
```

Once you have developed your own Java class files, you can change the directory and filename in jRun.txt to run your own program. Note that the program name is *Converter* and not *Converter.class* (the .class is assumed).

## 4.0 Writing or editing your own Java programs on the P800 or Psion

You will need a text editor. We have included our own *jText* program for this purpose.

## 4.1 Modifying the jRun.txt file

You can also use *jText* to modify the *jRun.txt* file to run whatever Java class file you want to. There are two points to note.

**Note 1:** *jText* cannot find files which are in the \system\ directory. Use *File Manager* to copy the *jRun.txt* file to another directory, edit it, then copy back to \system\apps\jRun\.

**Note 2:** the *jRun.txt* file must not end in a line feed character, otherwise it will not work. If you edit this file in *jText*, make sure there is no trailing line feed. If there is one, backspace until the file ends immediately after the Java class file name. Then save the file.

## 5.0 Developing Java programs with jCompile

Unless you are a superhuman Java programmer, programs that you develop yourself will not compile first time. For this reason, jCompile remembers the last file that you compiled. After correcting any error(s), when you run jCompile again select the "Last" option. This saves you having to repeatedly locate the same Java source file in the file dialog.

## 5.1 A note on Classpaths

When you have located your Java source file in the file dialog, jCompile will automatically add the current directory and its parent directory to the Java classpath. This enables both stand-alone source files and source files that are part of a Java package to be compiled.

## 6.0 Final note

I'm sorry but I can't help resolve general Java programming queries. I simply don't have the time. I will try to respond to any errors or problems specifically related to this software but, as stated above, you use it entirely at your own risk. Happy programming :-))

Malcolm Bryant
malcolm@freepoc.org
www.freepoc.org