Running Java Programs

Written by: Keith Fenske, http://www.psc-consulting.ca/fenske/
First version: Thursday, 10 January 2008
Document revised: Saturday, 13 February 2010
Copyright © 2008, 2010 by Keith Fenske. All rights reserved.

Introduction

The same Java program will run on many different operating systems: Linux, MacOS, Solaris, Windows, etc. This is an advantage to the programmer, because only one program needs to be written, and only one package needs to be distributed. This is an advantage to the user, because programs have a similar "look and feel" no matter where they are run.

Some systems come with Java pre-installed, for example, Apple Macintosh computers. For systems such as Microsoft Windows, you must download the Java "run-time environment" (JRE) first. The Java programming language was developed and is supported by Sun Microsystems, so that's the best place to get Java downloads:

> Sun Java Run-Time Environment (JRE for end users):
> http://www.java.com/getjava/
>
> Sun Java Software Development Kit (SDK for programmers):
> http://developers.sun.com/downloads/
>
> Sun Java NetBeans IDE (graphical compiler for programmers):
> http://www.netbeans.org/

The Java run-time environment gives you the ability to run the latest Java programs on your computer, including applets in most popular web browsers. The bigger software development kit (JDK or SDK) lets you compile and test Java programs. The even bigger "Integrated Development Environment" (IDE) is a better way of writing new Java programs. The IDE has at various times been known as "Forte for Java", "Sun ONE Studio", and now "NetBeans IDE".

Follow instructions on the Sun Java web site, with this advice. When given a choice between an on-line or "web" installation and an off-line or "full" installation, download the full setup and save the setup file in case you need it later. During setup, if you are given a choice between a typical or custom installation, choose the custom or complete setup with all optional components selected. The difference in size is small. Turn off automatic updates for Java, which can be found in the Control Panel, Java icon, Update tab for Windows.

Apple Macintosh

An easy way to download Java programs as ZIP files and to run them on Mac OS X is with the Apple Safari browser. Click on the web page's link for a ZIP file. Let Safari download and unpack the contents. The small Safari download window will have a button to show the files, or you can browse to the same location with the Mac Finder. Click on the file name for the main Java class, which is usually obvious or clearly identified in the PDF documentation. This will run the program with its default configuration. For anything more sophisticated, use a Terminal window similar to the "command prompt" in Windows or "shell" in Linux, and type a "java" command yourself with options.

Linux

Download and unpack the distribution or ZIP file into a directory/folder of your choice. Start a command shell and change directory (cd) to where the files are located. Type a "java" command with options as below. Note that Linux uses a forward slash (/) as a file separator instead of the backslash (\) in Windows.

Microsoft Windows

Windows 98 and Windows ME should be running the final release of Java 1.4.2, nothing older. The next version (Java 5.0) should work on Windows 98/ME but was tested mostly on Windows 2000 and Windows XP. Java 6.0 is the minimum for Windows Vista, and is optional for Windows 2000/XP. You can check your version of Java from the command line, also known as a console or "DOS" window. On Windows 98/ME, use Start button, Programs, MS-DOS Prompt. On Windows 2000/XP/Vista, use Start button, (All) Programs, Accessories, Command Prompt. Type this command, followed by the Enter key:

```
java  -version
```

If Java is installed, the response will be something like:

```
java version "1.5.0_17"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_17-b04)
Java HotSpot(TM) Client VM (build 1.5.0_17-b04, mixed mode, sharing)
```

And if Java is not installed, the response on Windows 2000/XP/Vista will be:

```
'java' is not recognized as an internal or external command,
operable program or batch file.
```

Windows 98/ME will just say "Bad command or file name". To close the command prompt (DOS window), type this command followed by the usual Enter key:

```
exit
```

Most small Java programs run from the command line. The same command line is what runs programs from desktop icons or with Start menu items. Larger packages may have their own installers to hide these details from you (known as *.EXE or *.MSI files on Windows), but you shouldn't be afraid of the command line.

Downloading Java Software

Java programs on this web site are in ZIP files. Support for ZIP "archives" is built into Windows XP/Vista and later. (Windows 2000 and earlier need separate ZIP software.) A description with links can be found on my free software page. Once you have downloaded and saved a ZIP file, in a folder that you can find again such as "My Documents", then you can show the contents of the ZIP file on Windows XP/Vista by double clicking on the file name in My Documents or Windows Explorer (opens up like a regular subfolder), or you can extract files by right clicking on the ZIP file name, left clicking on "Extract All", and following the prompts. You may need to do some cleaning up and/or moving of files to where you want them.

Put the program files into a folder (directory) on your hard drive. The name of the folder and the location are your choice, except it is easier if the name does not include spaces, because spaces are a problem on command lines unless you quote arguments (something you probably don't want to know about). Assume that files will go into a C:\JAVA folder, that is, a main-level folder called JAVA in the root directory of the C: disk drive, which is easy to find and use from DOS. Unpack the ZIP file into this folder. For the FileChecksum application, the program files look like this:

```
FileChecksum2.au  (3 KB, barking dog sound)
FileChecksum2.class  (17 KB, executable program)
FileChecksum2.doc  (32 KB, documentation in Microsoft Word format)
FileChecksum2.gif  (14 KB, sample program image)
FileChecksum2.ico  (7 KB, icon for Windows)
FileChecksum2.java  (46 KB, source code)
FileChecksum2.pdf  (70 KB, documentation in Adobe Acrobat format)
FileChecksum2User.class  (1 KB, helper class for main program)
GnuPublicLicense3.txt  (36 KB, legal notice)
RunJavaPrograms.pdf  (85 KB, these notes in Adobe Acrobat format)
```

You really only need to know the name of the main program (FileChecksum2) and where the files are located (C:\JAVA).

Running Class Files

Small Java programs are distributed as "class" files. One of the class files has the main program, usually the file with the simplest name that matches the name of the distribution package. For the FileChecksum example above, the main program is in a FileChecksum2.class file. To run this program on Windows, start a command prompt. Change to the folder with the program files and run the program with a "java" command:

```
c:
cd  \java
java  FileChecksum2
```

The first line switches to the C: disk drive, if not already there. The second line switches to the folder containing the Java program files. The third line runs the program. The program name "FileChecksum2" must appear exactly as shown; uppercase and lowercase letters are different in Java names. In this particular case, FileChecksum2 will open a graphical user interface or "GUI" window. When you are finished and close/exit the program, then you will return to the command prompt. You may close the command prompt with the "exit" command as shown above.

Running JAR Files

Larger Java applications are bundled in "JAR" files (meaning, Java archive). JAR files are really ZIP files with a manifest, if you want to rename them and look inside. They are used because major applications can have hundreds or thousands of class files — far too many to unzip into a regular directory. Assume that you download an application called "FluffySnakes3" in a JAR file. Then change the command line above to:

```
c:
cd  \java
java  -jar  FluffySnakes3.jar
```

The only possible difference from the previous example is a scary error message saying:

```
Failed to load Main-Class manifest attribute from FluffySnakes3.jar
```

which means that the creator of the JAR file didn't correctly identify the class file for the main program. Many systems will run a JAR file by clicking (or double clicking) on the JAR file name while viewing a directory in your file browser (i.e., Windows Explorer). Unfortunately,

some archive utilities associate themselves with the ".jar" suffix and prevent this from happening. The command line is the only guaranteed way of running a Java program.

Java Run-Time Options

For a list of run-time options for Java, use the following command line:

```
java  -help
```

The -X option shows additional, non-standard options. One of the most useful of these on Windows is -Xmx to increase the amount of memory available to an application. The default is around -Xmx60M or 60 megabytes (MB) for Java 1.4 on Windows 2000. Running an application with a command like:

```
java  -Xmx150M  FileChecksum2
```

would give the program a maximum of 150 megabytes of memory for temporary data. Please note that options to Java itself come after the "java" command and before the program name. Many Java programs have their own options after the program name.

The CLASSPATH Question

One complication may arise when trying to run a Java program. Java looks for an environment variable called CLASSPATH. If it finds this variable, then that is a list of folders where it looks for class files. It won't look anywhere else, not even in the current directory, unless the path contains "." as one of the choices. The symptom is an error message that says:

```
Exception in thread "main" java.lang.NoClassDefFoundError: FileChecksum2
```

To find out if your system has a CLASSPATH variable defined, type the following command in a DOS window:

```
set  CLASSPATH
```

To temporarily change the CLASSPATH variable to the current directory when running a program, use a command line similar to the following:

```
java  -cp  .  FileChecksum2
```

Remember to use the correct program name, of course! To permanently change the CLASS-PATH, you must find where it is being set. This may be in an old AUTOEXEC.* file in the root directory of your system disk (usually the C:\ folder), or it may be in Control Panel, System,

Advanced, Environment Variables on Windows 2000/XP. The CLASSPATH is actually a good idea; it's just different than the way Windows likes to do things. If you don't already have a CLASSPATH variable, then create one in Control Panel with the value:

`C:\JAVA;.`

assuming that the Java program files are in C:\JAVA as per the examples above. Then, no matter what your current directory is in a DOS window, Java will always look for Java program files in C:\JAVA first, then in the current directory (".") if it doesn't find them in C:\JAVA. Feel free to put the current directory before C:\JAVA:

`.;C:\JAVA`

If you already have a CLASSPATH from previous applications, insert C:\JAVA in the order that you want Java to search for program files, with semicolons (;) between folder names. This will make desktop icons and Start menu items much easier (next section).

Desktop Icons and Menu Shortcuts

Should you find a Java program to be popular, you can create a Start menu item or desktop shortcut on Windows 2000/XP/Vista. This works well for graphical applications; it does not apply to applications that run in "console" mode (command-line interface only).

For a desktop icon, right click anywhere on the background (not on a program window), left click on New, then left click on Shortcut. For a Start menu item, right click on the Start button (bottom left-hand corner), left click on Explore (applies to this user only) or Explore All Users. Then in any folder of "Start Menu", right click on the background, left click on New, then Shortcut. You will first be asked to type the location of the item. Type a partial Java command like:

`java.exe  FileChecksum2`

Then click the Next button. You will be asked to type a name for this shortcut. You may type any name that you want. Then click the Finish button. Windows will create the desktop icon or menu shortcut. You can review the settings by right clicking on the icon or name, and left clicking on Properties. You will notice that Windows has expanded the partial command line:

`%windir%\system32\java.exe  FileChecksum2`

with the program starting in a folder called %windir% (a substitution for C:\WINDOWS on most Windows XP computers). If you have a CLASSPATH variable defined (as recommended above), the starting folder isn't important. If you don't have a CLASSPATH, then the starting

folder must be the location of the Java class files. If there is a mistake when you try to run the icon or the menu item, you will briefly see a black DOS window, which immediately closes after printing an error message that you don't have time to read.

If you don't want to see a DOS console window while the Java program is running, then choose the "Run: Minimized" option on the Shortcut dialog tab. An additional property is the "Change Icon" button. Use this to set a specific program icon, such as an icon that came with the program in an *.ICO file.

Removal or Uninstall

For Java programs on this web site, remove the programs from your computer by deleting the installation files. If the folder that contained the files is now empty, you may also delete the folder ... if you created the folder, of course, not the system. If you created desktop shortcuts or Start menu items, then delete those too. Unless otherwise said, there are no configuration or preference files, and no information is stored in the Windows system registry. You don't need an "uninstall" program.

Sun Solaris

See Linux for general instructions. Most Java options are the same as described above for Windows.

file: RunJavaPrograms.doc 2010-02-13