Java Application: FileSearch

Written by: Keith Fenske, http://kwfenske.github.io/
First version: Saturday, 29 September 2007
Document revised: Saturday, 13 February 2010
Copyright © 2007 by Keith Fenske. Apache License or GNU General Public License.

Description

FileSearch is a Java 1.4 application to find files that contain (or don't contain) a given string. The string may be in plain text or it may be a Java regular expression. Such a trivial search should be part of the operating system, and in fact, once was. As bigger and more impressive features were added to Windows, it lost the ability to search files for arbitrary bytes of text. Windows 98/ME/2000 could find words buried in files with unknown formats; Windows XP/Vista/7 will search only supported file types.

A regular expression is a way of specifying relationships between elements of a complex pattern. You don't need to understand regular expressions to use this program. Please refer to the following on-line sources for information about Java regular expressions and for the related topic of character set encodings:

> Java Class: Pattern
> http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html
>
> Alan Wood's Unicode Resources
> http://www.alanwood.net/unicode/

You can safely ignore the option for regular expressions, as they are turned off by default. An attempt is made to break input files into lines, before applying a search pattern. However, you should not rely on this for files with very long lines (thousands of characters) or with large amounts of binary data. Compressed data is not expanded for ZIP files or other archive types.

Apache License or GNU General Public License

FileSearch is free software and has been released under the terms and conditions of the Apache License (version 2.0 or later) and/or the GNU General Public License (GPL, version 2 or later). This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the license(s) for more details. You should have received a copy of the licenses along with this program. If not, see the http://www.apache.org/licenses/ and http://www.gnu.org/licenses/ web pages.

Installation

You must have the Java run-time environment (JRE) installed on your computer. FileSearch was developed with Java 1.4 and should run on later versions. It may also run on earlier versions, but this has not been tested. You can download the JRE from Oracle (formerly Sun Microsystems):

> JRE for end users: http://www.java.com/getjava/
> SDK for programmers: http://www.oracle.com/technetwork/java/
> IDE for programmers: http://www.netbeans.org/

Once Java is installed, you need to put the program files for FileSearch 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. Assume that files will go into a "C:\Java" folder. Then create the folder and unpack the Java *.class files into this folder (if you received the program as a ZIP file). The files look something like this:

> ApacheLicense20.txt (12 KB, legal notice)
> FileSearch1.class (24 KB, executable program)
> FileSearch1.doc (33 KB, this documentation in Microsoft Word format)
> FileSearch1.gif (20 KB, sample program image)
> FileSearch1.ico (87 KB, icon for Windows)
> FileSearch1.jar (14 KB, archive file with same class files inside)
> FileSearch1.java (72 KB, source code)
> FileSearch1.manifest (1 KB, main class manifest for archive file)
> FileSearch1.pdf (73 KB, this documentation in Adobe Acrobat format)
> FileSearch1User.class (1 KB, helper class for main program)
> GnuPublicLicense3.txt (35 KB, legal notice)
> RunJavaPrograms.pdf (60 KB, more notes about running Java)
> ShowCharSets2.java (3 KB, program to generate Java character set names)
> ShowCharSets2.txt (19 KB, sample file with Java character set names)

To run the program on Windows, start a DOS command prompt, which is Start button, Programs, Accessories, Command Prompt on Windows XP/Vista/7. Change to the folder with the program files and run the program with a "java" command:

```
c:
cd \java
java  FileSearch1
```

The program name "FileSearch1" must appear exactly as shown; uppercase and lowercase letters are different in Java names. Some systems (Macintosh) will run a main "class" file by clicking on the class file name while viewing a directory in the file browser (Mac Finder). Many systems

will run a "jar" file by clicking (or double clicking) on the jar file name (Windows Explorer). The command line is the only guaranteed way of running a Java program. Should you find this program to be popular, you can create a Start menu item or desktop shortcut on Windows XP/Vista/7 with a target of "java.exe FileSearch1" starting in the "C:\Java" folder.

One complication may arise when trying to run this 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: FileSearch1
```

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, use the following command line:

```
java -cp . FileSearch1
```

To permanently change the CLASSPATH, you must find where it is being set. This is in Control Panel, System, Advanced, Environment Variables on Windows XP/Vista/7.

Removal or Uninstall

To remove this program from your computer, delete the installation files listed above. 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. There are no hidden configuration or preference files, and no information is stored in the Windows system registry. You don't need an "uninstall" program.

Graphical Versus Console Application

The Java command line may contain options or file and folder names. If no file or folder names are given on the command line, then this program runs as a graphical or "GUI" application with the usual dialog boxes and windows. See the "-?" option for a help summary:

```
java FileSearch1 -?
```

The command line has more options than are visible in the graphical interface. An option such as -u14 or -u16 is recommended because the default Java font is too small. If file or folder

names are given on the command line, then this program runs as a console application without a graphical interface. A generated report is written on standard output, and may be redirected with the ">" or "1>" operators. (Standard error may be redirected with the "2>" operator.) An example command line is:

```
java FileSearch1 -s "fluffy snakes" d:\documents >report.txt
```

The console application will return an exit status equal to the number of files reported, whether found or not found. See the -m option on the command line. The graphical interface can be very slow when the output text area gets too big, which will happen if thousands of files are reported.

Restrictions and Limitations

The speed of this program depends upon the speed of your computer's hardware (of course) and the complexity of the search string. When searching for plain ASCII text or Unicode characters from 0x20 to 0x7E, the "(raw data bytes)" encoding is about 40% faster than the local system's "(default encoding)". Uppercase and lowercase letters are normally considered to be equal; selecting the "case" option (-c) avoids this comparison and almost doubles the speed. Even an old Intel Pentium 4 processor at 3.0 GHz should be able to scan large files at 15 megabytes per second (MB/s) as raw data bytes with the "case" option enabled. The "nulls" option (-n) to ignore <NUL> and <DEL> control characters is not for performance, but is useful when plain text may appear as 7-bit ASCII or 16-bit Unicode. Don't use the "(raw data bytes)" encoding with 8-bit characters (or higher) unless you fully understand character sets.