Java Application: FontNames

Written by: Keith Fenske, http://www.psc-consulting.ca/fenske/
First version: Sunday, 11 March 2007
Document revised: Saturday, 13 February 2010

Description

FontNames is a Java 1.4 application to extract font names from OpenType and TrueType font files. Each font can have more than one name in different languages, and TrueType collections can contain more than one font. Only Unicode names are recognized; most other country or language specific names are reported in US-ASCII. For information on the format of font files, start with the following on-line references:

Microsoft TrueType Font Properties Extension
http://www.microsoft.com/typography/TrueTypeProperty21.mspx

The OpenType Font File
http://www.microsoft.com/typography/otspec/otff.htm
http://www.microsoft.com/typography/otspec/name.htm
http://www.microsoft.com/typography/otspec/os2.htm

The TrueType Font File
http://developer.apple.com/textfonts/TTRefMan/RM06/Chap6.html
http://developer.apple.com/textfonts/TTRefMan/RM06/Chap6name.html
http://developer.apple.com/textfonts/TTRefMan/RM06/Chap6OS2.html

OpenType (*.OTF) and TrueType (*.TTC, *.TTF) files are supported. Adobe PostScript (*.PFB, *.PFM) files are not supported.

There aren't many options. You can choose some of the optional name fields (copyright, version, etc). Then open one or more font files with the "Open" button. A summary will be shown in a scrolling text window. You may save the results to a file with the "Save Output" button, but the output file will be in your system's default character set. If the display has characters from other languages such as Chinese or Eastern European, then it is better to copy and paste the text directly into a Unicode-aware application like Microsoft Word. This program works best if you have the "Arial Unicode MS" font installed.

To rename font files using their internal names, see the FontRename Java application.

GNU General Public License (GPL)

FontNames is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License or (at your option) any later version. 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 GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see the http://www.gnu.org/licenses/ web page.

Name Fields Reported

FontNames reports several strings from the "name" table and one field from the "OS/2" table:

- always reported: font family name (name ID #1);
- optional detailed names: font subfamily (name ID #2), unique subfamily (#3), font full name (#4), PostScript name (#6), preferred family name (#16), preferred subfamily (#17), compatible name (#18), PostScript CID "findfont" name (#20);
- optional designer information: vendor (name ID #8), designer (#9), vendor URL (#11), designer URL (#12), OS/2 vendor ID;
- optional legal information: copyright (name ID #0), trademark (#7), license (#13), license URL (#14); and
- optional miscellaneous information: version (name ID #5), description (#10), sample text (#19), plus any unknown name ID numbers.

These categories are somewhat arbitrary and may be changed in the future.

Installation

You must have the Java run-time environment (JRE) installed on your computer. FontNames 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. For Macintosh computers, the version of Java is determined by your version of MacOS. For Windows, Linux, and Solaris, you can download the JRE from Sun Microsystems:

Sun Java
JRE for end users: http://www.java.com/getjava/
SDK for programmers: http://developers.sun.com/downloads/
IDE for programmers: http://www.netbeans.org/

Once Java is installed, you need to put the program files for FontNames 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:

```
FontNames3.class  (50 KB, executable program)
FontNames3.doc  (37 KB, this documentation in Microsoft Word format)
FontNames3.gif  (14 KB, sample program image)
FontNames3.ico  (7 KB, icon for Windows)
FontNames3.jar  (27 KB, archive file with same class files inside)
FontNames3.java  (132 KB, source code)
FontNames3.manifest  (1 KB, main class manifest for archive file)
FontNames3.pdf  (77 KB, this documentation in Adobe Acrobat format)
FontNames3Filter.class  (1 KB, helper class for main program)
FontNames3User.class  (1 KB)
GnuPublicLicense3.txt  (35 KB, legal notice)
RunJavaPrograms.pdf  (88 KB, more notes about running Java)
```

To run the program on Windows, start a DOS command prompt, which is Start button, Programs, Accessories, Command Prompt on Windows 2000/XP. Change to the folder with the program files and run the program with a "java" command:

```
c:
cd  \java
java  FontNames3
```

The program name "FontNames3" 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 2000/XP with a target of "java.exe FontNames3" 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: FontNames3
```

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 . FontNames3
```

(Thank you to "Friendly" in alt.binaries.fonts for this -cp suggestion.) To permanently change the CLASSPATH, 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.

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 FontNames3 -?
```

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.) The report will be in your system's default character set. An example command line is:

```
java FontNames3 -s d:\fonts >report.txt
```

The console application will return an exit status of 1 for success, –1 for failure, and 0 for unknown. 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

Not all font files are correctly structured. Before reporting an error in this program, make sure that the error isn't in the font file. Frequent mistakes by font vendors are incorrect character set encodings, missing name tables, etc. It is known that this program lacks the ability to convert some Macintosh character sets to Unicode.

---

file: FontNames3.doc 2010-02-13