Java Application: FontChecksum

Written by: Keith Fenske, http://www.psc-consulting.ca/fenske/
First version: Wednesday, 7 March 2007
Document revised: Saturday, 13 February 2010

Description

FontChecksum is a Java 1.4 application to verify the internal checksums in OpenType and TrueType font files. Sometimes when you download a file, or copy a file from a worn CD/DVD, what you receive differs from the original. After downloading or copying twice, you may have two files with the same size, the same date, and the same apparent contents. When compared byte-by-byte with commands such as “comp” on DOS/Windows or “cmp” on UNIX, only a few bits or bytes are different. The rest of the files are the same. Your question is which file is correct. One of them must be wrong. (Both could be wrong!) Since font files contain internal checksums, you can use this information to test which of the files is more correct. Note that freeware and shareware fonts produced with older font tools often have numerous checksum errors.

All proper OpenType and TrueType font files have a file checksum value of 0xB1B0AFBA, calculated by adding the file as 32-bit words and ignoring overflow. This is done originally by first setting a word called “checkSumAdjustment” in the “head” table to zero, then summing the file as words, then setting the adjustment to 0xB1B0AFBA minus the zero-based sum. Since all correct OTF and TTF font files have this same external checksum, it should not be necessary to understand the internal format of the files. Unfortunately, TrueType collection (TTC) files do not sum to 0xB1B0AFBA. This may have to do with the tools used to assemble TTC files. Being much bigger than OTF or TTF files, TTC files are more likely to suffer data corruption. To test OTF, TTC, and TTF font files, this program reads all “table directory” entries in each “Offset Table” for each internal font. (OTF and TTF files have one internal font; TTC files have more than one.) The overall file checksum is calculated for OTF and TTF files, but not for TTC files. If all tested checksums are correct, then the total file is assumed to be correct. 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/head.htm

The TrueType Font File
http://developer.apple.com/textfonts/TTRefMan/RM06/Chap6.html
http://developer.apple.com/textfonts/TTRefMan/RM06/Chap6head.html

See the CompareFolders application for comparing two folders to determine if all files and subfolders are identical. See the FileChecksum application to generate or test checksums for a single file. See the FindDupFiles application to look for duplicate files based on MD5 checksums.

GNU General Public License (GPL)

FontChecksum 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.

Installation

You must have the Java run-time environment (JRE) installed on your computer. FontChecksum 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 FontChecksum 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:

```
FontChecksum2.class  (23 KB, executable program)
FontChecksum2.doc  (36 KB, this documentation in Microsoft Word format)
FontChecksum2.gif  (20 KB, sample program image)
FontChecksum2.ico  (8 KB, icon for Windows)
```

```
FontChecksum2.jar  (14 KB, archive file with same class files inside)
FontChecksum2.java  (78 KB, source code)
FontChecksum2.manifest  (1 KB, main class manifest for archive file)
FontChecksum2.pdf  (75 KB, this documentation in Adobe Acrobat format)
FontChecksum2Filter.class  (1 KB, helper class for main program)
FontChecksum2User.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  FontChecksum2
```

The program name "FontChecksum2" 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 FontChecksum2" 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: FontChecksum2
```

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  .  FontChecksum2
```

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 FontChecksum2 -?
```

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 FontChecksum2 -s d:\fonts >errors.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.

---

file: FontChecksum2.doc 2010-02-13