Java Application: EraseDisk

Written by: Keith Fenske, http://kwfenske.github.io/
First version: Wednesday, 23 March 2022
Document revised: Monday, 28 March 2022
Copyright © 2022 by Keith Fenske. Apache License or GNU General Public License.

Description

EraseDisk is a Java 1.4 graphical (GUI) application to erase and test disk drives or flash drives. Large temporary files are created and filled with zeros, ones, or pseudo-random data. Previously deleted files are overwritten. Existing files are not affected. This cleans up an old disk before it goes in a new location. Don't trust a new disk until you write data, then read to confirm. One complete test is usually enough. (Repeated testing may degrade flash drives.) Use this program as follows:

1. If you want to erase an entire disk drive or partition (including all files), then first "format" or "initialize" the drive according to your system's usual procedure.

2. Otherwise, empty the "Recycle Bin" or "Trash" folder. This releases hidden space still allocated to some deleted files.

3. Run this program.

4. Navigate to any folder on the disk drive where temporary files can be created. Files with names similar to "ERASE123.DAT" are assumed to belong to this program and will be replaced or deleted without notice.

5. Decide whether you want the program to write only zeros (one pass) or to write three passes with all ones (0xFF bytes), pseudo-random bytes, then all zeros (0x00 bytes).

6. Decide whether you want the program to read verify the written data to test if the drive is working correctly. Skip this step if you intend to destroy the disk drive after it is erased.

7. Click the "Start" button and be patient.

8. Check that the number of bytes reported by this program agrees with what your system says for free space. (Java doesn't really know.) Also, Java doesn't report hardware errors if the operating system recovers. Look at your system error logs; see Event Viewer on Microsoft Windows.

EraseDisk is not a "secure erase" program and does not meet the DoD 5220.22-M standard for the United States. Please refer to the following Wikipedia.org web page:

http://en.wikipedia.org/wiki/Data_remanence

There are many such applications available, some for free, by searching "disk wipe" or "secure erase" on Google.com or similar web search engines. The programs that do the best job operate at a low level, requiring direct or "raw" access to a disk drive (not allowed for regular users), or running as stand-alone programs without a full operating system. They are sensitive to what hardware is supported and often fail when new hardware requires drivers that are not included in their programming. Other programs (such as this one) work within a standard file system, which allows them to be run by regular users, but prevents them from erasing any part of a drive occupied by file directories or boot blocks. Most people just want to make sure that files they deleted are truly gone, and this program is good enough for that.

Apache License or GNU General Public License

EraseDisk 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.

Frequently Asked Questions (FAQ)

Why is there no read verify when writing zeros or ones? This feature is currently disabled. Any sequence that repeats the same value can be highly manipulated. Since all zeros are the same, there is no guarantee that data is coming from the disk drive, or even the correct location.

What's so great about pseudo-random data? It can't be compressed or predicted by the operating system or hardware. The system has no choice but to do what it's told. Generating random numbers is slower. The most time-consuming part is actually the read verify.

I am using disk compression and this program never ends. Of course not, not when writing ones or zeros or multiple copies of the same byte. Instead of writing a constant value, the system counts the repetitions until it reaches a huge number like 9,223,372,036,854,775,807 (the largest signed 64-bit integer). Use pseudo-random data with compressed disk drives or folders.

Why doesn't the program know the exact amount of free space on a disk? Compression and encryption affect the reported space. Only when the program runs out of space does it know that the disk drive is full ... probably.

Can this program be changed to meet the "secure erase" standard? No. First, there are multiple standards for securely erasing a disk drive. Many programs that claim this standard in fact ignore special cases which prevent them from finishing the job. Second, the proper way to

erase a disk drive is to overwrite every data block on the entire disk, including the boot block, disk directories, file contents, and bad (redirected) sectors. This can't be done from within an existing file system.

Does the program support custom data patterns? Yes, after some editing and compilation. Each write pass can use any byte from 0x00 to 0xFF, or pseudo-random data. Each pass can have a read verify. Each read verify can prompt the user to eject and reinsert removable media before the verify begins (for flash drives, floppy disks, etc). There can be any number of passes.

Installation

You must have the Java run-time environment (JRE) installed on your computer. EraseDisk 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/download/
SDK for programmers: http://www.oracle.com/java/ or the OpenJDK builds
IDE for programmers: http://www.eclipse.org/ or http://www.netbeans.org/

Once Java is installed, you need to put the program files for EraseDisk 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)
EraseDisk3.class (34 KB, executable program)
EraseDisk3.doc (36 KB, this documentation in Microsoft Word format)
EraseDisk3.ico (87 KB, icon for Windows)
EraseDisk3.jar (21 KB, archive file with same class files inside)
EraseDisk3.java (129 KB, source code)
EraseDisk3.manifest (1 KB, main class manifest for archive file)
EraseDisk3.pdf (79 KB, this documentation in Adobe Acrobat format)
EraseDisk3.png (17 KB, sample program image)
EraseDisk3Grid.class (2 KB, helper class for main program)
EraseDisk3User.class (1 KB)
GnuPublicLicense3.txt (35 KB, legal notice)
RunJavaPrograms.pdf (60 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 XP/Vista/7. Change to the folder with the program files and run the program with a "java" command:

```
c:
cd \java
java  EraseDisk3
```

The program name "EraseDisk3" 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 EraseDisk3" 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: EraseDisk3
```

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  .  EraseDisk3
```

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 for the position and size of the application window, and the size of the display font. See the "-?" option for a help summary:

```
java EraseDisk3 -?
```

The command line has more options than are visible in the graphical interface. An option such as -u16 or -u18 is recommended because the default Java font is too small.

Restrictions and Limitations

As computers become more protective of their files, it is increasingly difficult to be certain that a particular file has actually been deleted. (See any feature that offers to restore previous versions of a file.) Devices with small or unusual block sizes may not be completely erased. Some systems leave partial file names in old directory entries even after files are deleted. Others store very small files inside the directory structure (around 728 bytes or less for NTFS). The only practical way of removing this information is to reformat the disk drive or partition ... and lose all files. An operating system's disk cache may make a read verify meaningless if the amount of data written to the disk is less than the physical memory size (RAM) on a computer. Windows 2000/XP/Vista/7 tends to misallocate a few clusters when large FAT32 volumes are nearly full or files reach their maximum size; these show up later as "lost" single-cluster files in CHKDSK.