DiscDataWipe (tm) Kenneth Ives kenaso@tx.rr.com

I am open to ways to improve this application, please email me.

Visual Basic 6.0 with Service Pack 6 runtime files required.
To obtain required files (VBRun60sp6.exe):
http://www.microsoft.com/downloads/details.aspx?FamilyId=7B9BA261-7A9C-43E7-9117-F6730
77FFB3C

VBRun60sp6.exe installs Visual Basic 6.0 SP6 run-time files.
http://support.microsoft.com/kb/290887

This software has been tested on Windows XP through Windows 7.
Windows 9x, 2000 and NT4 are no longer supported.

This application can process files in excess of 2gb.

=
*** WARNING *** WARNING *** WARNING *** WARNING *** WARNING *** WARNING ***
=

You acknowledge that this software is subject to the export control laws and regulations of the United States ("U.S.") and agree to abide by those laws and regulations. Under U.S. law, this software may not be downloaded or otherwise exported, reexported, or transferred to restricted countries, restricted end-users, or for restricted end-uses. The U.S. currently has embargo restrictions against Cuba, Iran, Iraq, Libya, North Korea, Sudan, and Syria. The lists of restricted end-users are maintained on the U.S. Commerce Department's Denied Persons List, the Commerce Department's Entity List, the Commerce Department's List of Unverified Persons, and the U.S. Treasury Department's List of Specially Designated Nationals and Blocked Persons. In addition, this software may not be downloaded or otherwise exported, reexported, or transferred to an end-user engaged in activities related to weapons of mass destruction.

=

REFERENCE:

NIST (National Institute of Standards and Technology)
FIPS (Federal Information Processing Standards Publication)
SP (Special Publications)
http://csrc.nist.gov/publications/PubsFIPS.html

FIPS 180-2 (Federal Information Processing Standards Publication)
dated 1-Aug-2002, with Change Notice 1, dated 25-Feb-2004
http://csrc.nist.gov/publications/fips/fips180-2/FIPS180-2_changenotice.pdf

FIPS 180-3 (Federal Information Processing Standards Publication)
dated Oct-2008 (supercedes FIPS 180-2)
http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf

FIPS 180-4 (Federal Information Processing Standards Publication)
dated Mar-2012 (Supercedes FIPS-180-3)
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

Examples of SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 are available at
http://csrc.nist.gov/groups/ST/toolkit/examples.html

Guidelines for Media Sanitization (SP800-88)

http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf

Feb-2009 NIST announces the release of Special Publication 800-106, Randomized Hashing for Digital Signatures. This recommendation provides a technique to randomize the input messages to hash functions prior to the generation of digital signatures to strengthen security of the digital signatures.
http://csrc.nist.gov/publications/nistpubs/800-106/NIST-SP-800-106.pdf

MD4, MD5, RIPEMD Algorithms have been compromised at the rump session of Crypto 2004. It was announced that Xiaoyun Wang, Dengguo Feng, Xuejia Lai and Hongbo Yu found collisions for MD4, MD5, RIPEMD, and the 128-bit version of HAVAL.
http://eprint.iacr.org/2004/199.pdf

Feb-2005 SHA-1 has been compromised. Recommended that you do not use for password or document authentication.
http://www.schneier.com/blog/archives/2005/02/sha1_broken.html
http://csrc.nist.gov/groups/ST/toolkit/documents/shs/NISTHashComments-final.pdf

Mar-2005 Demonstrating a technique for finding MD5 collisions quickly. Eight hours on 1.6 GHz computer.
http://cryptography.hyperlink.cz/md5/MD5_collisions.pdf

Jun-2005 Two researchers from the Institute for Cryptology and IT-Security have generated PostScript files with identical MD5-sums but entirely different (but meaningful!) content.
http://www.schneier.com/blog/archives/2005/06/more_md5_collis.html

March 15, 2006: The SHA-2 family of hash functions (i.e., SHA-224, SHA-256, SHA-384 and SHA-512) may be used by Federal agencies for all applications using secure hash algorithms. Federal agencies should stop using SHA-1 for digital signatures, digital time stamping and other applications that require collision resistance as soon as practical, and must use the SHA-2 family of hash functions for these applications after 2010. After 2010, Federal agencies may use SHA-1 only for the following applications:

- hash-based message authentication codes (HMACs)
- key derivation functions (KDFs)
- random number generators (RNGs)

Regardless of use, NIST encourages application and protocol designers to use the SHA-2 family of hash functions for all new applications and protocols.
http://csrc.nist.gov/groups/ST/hash/policy.html

****

****
** Hard Drive Disposal **
****

If the hard disk that has had data classified greater than "CONFIDENTIAL", then the disk should be replaced with a new one. Since the cost of a fixed disk has dropped so dramatically, this should not be a factor. You should be considering the question, "What is my information worth to someone else?".

Steps to follow to dispose of the old hard drive:

1. Overwrite multiple times with random data (Min 5 times). I recommend the Dban web site and creating a bootable CD or USB device that will wipe every sector on a disk. This is freeware and several governments approve its use.

http://www.dban.org/
2. Remove disk from the old desktop or laptop and record the manufacturer, model, serial number, date of destruction and name of individual performing this process.
3. Plate area should be drilled in several places using a 1/2 inch drill bit.
4. Disintegrate, incinerate, pulverize, shred, or melt the hard drive.

All of the above should be witnessed by at least two additional persons and documented.

Ref: Guidelines for Media Sanitization (SP800-88)
http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf

****

For more information regarding clearing and sanitizing security standard:

References:
National Institute of Standards and Technologies Publications
http://csrc.nist.gov/publications/PubsSPs.html

Guidelines for Media Sanitization
http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf

Generally Accepted Principles and Practices for Securing Information Technology Systems
http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf

National Industrial Security Program Operating Manual (NISPOM)
DoD 5220.22-M, dtd February 28, 2006.
Chapter 8 -Information System Security, Section 3-Common Requirements, 8-301 Clearing and Sanitization
http://www.dss.mil/isp/fac_clear/download_nispom.html

San Diego Industrial Security Awareness Council (ISAC)
http://www.sdisac.com/
http://www.sdisac.com/clearing_and_sanitization_matrix.doc

US Department of Defense in the clearing and sanitizing standard DoD 5220.22-M recommends the approach "Overwrite all addressable locations with a character, its complement, then a random character and verify" (see table with comments) for clearing and sanitizing information on a writable media.

US Department of Defense 5220.22-M Clearing and Sanitization Matrix

Media	Clear	Sanitize
Magnetic Tape1		
Type I	a or b	a, b, or m
Type II	a or b	b or m
Type III	a or b	m
Magnetic Disk		
Bernoullis	a, b, or c	m
Floppies	a, b, or c	m
Non-Removable Rigid Disk	c	a, b, d , or m
Removabel Rigid Disk	a, b, or c	a, b, d , or m
Optical Disk		
Read Many, Write Many	c	m

Read Only		m, n
Write Once, Read Many (Worm)		m, n
Memory		
Dynamic Random Access memory (DRAM)	c or g	c, g, or m
Electronically Alterable PROM (EAPROM)	i	j or m
Electronically Erasabel PROM (EEPROM)	i	h or m
Erasable Programmable (ROM (EPROM)	k	l, then c, or m
Flash EPROM (FEPROM)	i	c then i, or m
Programmable ROM (PROM)	c	m
Magnetic Bubble Memory	c	a, b, c, or m
Magnetic Core Memory	c	a, b, e, or m
Magnetic Plated Wire	c	c and f, or m
Magnetic Resistive Memory	c	m
Nonvolatile RAM (NOVRAM)	c or g	c, g, or m
Read Only Memory ROM	m	
Static Random Access Memory (SRAM)	c or g	c and f, g, or m
Equipment		
Cahtode Ray Tube (CRT)	g	q
Printers		
Impact	g	p then g
Laser	g	o then g

US Department of Defense 5220.22-M Clearing and Sanitization Matrix

a. Degauss with a Type I degausser
b. Degauss with a Type II degausser
c. Overwrite all addressable locations with a single character
d. THIS METHOD NOT APPROVED FOR SANITIZING MEDIA THAT CONTAINS
   TOP SECRET INFORMATION.
   1. Before any sanitization product is acquired, careful analysis to the overall costs associated with overwrite/sanitization should be made. Depending on the contractor's environment, the size of the drive and the differences in the individual products time to perform the sanitization, destruction of the media might be the preferred (i.e., economical) sanitization method.
   2. Overwrite all addressable locations with a character, then its complement. Verify "complement" character was written successfully to all addressable locations, then overwrite all addressable locations with random characters; or verify third overwrite of random characters. Overwrite utility must write/read to "growth" defect list/sectors or disk must be mapped before initial classified use and remapped before sanitization. Difference in the comparison lists must be discussed with the DSS Industrial Security Representative (IS Rep) and/or Information System Security Professional (ISSP) before declassification. Note: Overwrite utilities must be authorized by DSS before use.
e. Overwrite all addressable locations with a character, its
   complement, then a random character
f. Each overwrite must reside in memory for a period longer than the
   classified data resided
g. Remove all power to include battery power
h. Overwrite all locations with a random pattern, all locations with
   binary zeros, all locations with binary ones
i. Perform a full chip erase as per manufacturer's data sheets
j. Perform i above, then c above, a total of three times
k. Perform an ultraviolet erase according to manufacturer's
   recommendation
l. Perform k above, but increase time by a factor of three.
m. Destroy - Disintegrate, incinerate, pulverize, shred, or melt.
n. Destruction required only if classified information is contained.

o. Run five pages of unclassified text (font test acceptable).
p. Ribbons must be destroyed. Platens must be cleaned.
q. Inspect and/or test screen surface for evidence of burned-in
   information. If present, the cathode ray tube must be destroyed.

****

****
** Overview
****

When encountering double byte characters, like French or one of the oriental languages, and DiscDataWipe did not successfully remove the file or folder then follow these steps:

1. Close DiscDataWipe
2. Open Windows Explorer and navigate to the offending folder
3. Manually delete the file or folder
4. Empty the Recycle Bin
5. Restart DiscDataWipe
6. Wipe the free space of this particular drive

Busting the Biggest PC Myths
http://pcworld.about.com/magazine/2208p107id116572.htm

"Want to erase data from a hard drive you plan to toss? Don't bother with a magnet. Overwrite the data that is stored on the media instead. For a flash drive, fill up the drive with anything, like pictures of your beloved dachshund. Unlike with magnetic media, from which experts can usually recover at least some overwritten data, once new data is written to flash media, the old data is gone forever."

Note from Ken:

> If you want to clean a flash drive, it is much faster to use Windows Explorer to delete all the files and folders first. When finished, select the DiscDataWipe application to wipe the free space on the flash drive.

Good information from Alfred Hellmüller:

> Disk Defragmenters as well as Wipers works with numerous repeating write operations. Flash memories like USB Sticks or SD Cards do have a limited life of about 10.000 to 10e6 (high reliability) erase and write operations. Even read operations are degrading the data quality response. Increasing Bit failures.
>
> Some areas of a Flash are extremely exposed such as those that holds index structures of the file system. The 10'000 operations can be reached in a short time because any write operation includes a delete operation in advance.
>
> Conclusion: We should avoid any Write operations on USB sticks whenever possible. Both, Defragmenter and Wiper applications are high grade toxic Procedures for Flash memories.
>
> References:
> Storage Search.com
> http://www.storagesearch.com/reliability.html

Maximizing Performance and Reliability in Flash Memory Devices
by Randy Martin, QNX Software Systems
Ecnmag.com - March 01, 2006
http://www.ecnmag.com/maximizing-performance-and-reliability.aspx?menuid=580

Google: flash memories write operations limit

Size definitions used by various disk manufacturers

Bit	0 or 1
Nibble	4 Bits
Byte	8 Bits
Kibibit	1,024 bits
Kilobit	1,000 bits
Kibibyte	1,024 bytes
Kilobyte	1,000 bytes
Mebibit	1,048,576 bits
Megabit	1,000,000 bits
Mebibyte	1,048,576 bytes
Megabyte	1,000,000 bytes
Gibibit	1,073,741,824 bits
Gigabit	1,000,000,000 bits
Gibibyte	1,073,741,824 bytes
Gigabyte	1,000,000,000 bytes
Tebibit	1,099,511,627,776 bits
Terabit	1,000,000,000,000 bits
Tebibyte	1,099,511,627,776 bytes
Terabyte	1,000,000,000,000 bytes
Pebibit	1,125,899,906,842,624 bits
Petabit	1,000,000,000,000,000 bits
Pebibyte	1,125,899,906,842,624 bytes
Petabyte	1,000,000,000,000,000 bytes

In 1998 the IEC changed it's measurements so that what you consider a gigabyte (1024 mb or 2^30) was renamed a gibibyte. A gibibyte is now formally recognised as 1000 mb, even though no operating system (like windows) use this definition.

Hard drives use the term "gigabytes" which would be what the 40GB stands for. But, Microsoft uses a different way to measure gigabytes (they're actually gibibytes) so that's why you "lost" some 3.8GB from your hard drive. It happens to everyone. It's just the conflict of two different numbering systems.

The difference between those two numbering systems is seven percent. So, when you buy a 40 GB drive, Windows sees it as 37.2 GB because:

    40 GB - 7% = 37.2 GB.

Begin processing the request to either wipe a file, files within a folder, or a complete folder and it's sub-folders.

The process used to wipe a file follows this scenario:

1. Open the file as binary write
2. Overwrite the contents of the file 1-99 times with the wiping pattern selected. See the Wiping Patterns below.
3. Close the file. It's contents have been overwritten.

4. Rename the folder or file 26 times. Beginning with the letter "A"
   and ending with the letter "Z".

   Example: 1. Foo.txt --> AAA.AAA
             2. AAA.AAA --> BBB.BBB
                 ...
            26. YYY.YYY --> ZZZ.ZZZ

5. The folder or file date properties are updated with a random generated
   new timestamp. Date properties include date created, date last accessed,
   date last modified.

6. The file size is adjusted to zero bytes. (Opened as output and closed)

7. The folder or file is now deleted.

Whenever you perform a DiscDataWipe on a sensitive document(s) or a number of files, you should wipe the free space when you are finsihed. This removes any fragments that may have been left behind. When finished wiping the free space, perform a defrag of the drive and reboot the PC. This will purge most deleted entries while making the files be of a more contiguous nature for faster loading and execution. This process will also re-verify the pointers of all existant files and folders in the VTOC and FAT.

****
Wiping Patterns
****
All writes to the drive are based on the sector size. One pass means one complete overwrite from beginning to end. A pass can be performed 1-99 times based on user selection.

****
** Wipe free space **
****
Creates a temp folder in the temp directory of the drive to be cleaned, if one does not exist. Within this folder, a series of hidden files are created and filled with specific values. Maximum size of each file is 2GB. When finished, the folder and files within are emptied and deleted.

If using binary zeroes (0x00), the drive will be overwriten three times (three passes).

if using the alternate method, three overwrites will be performed which is equal to one pass. Each of the temporary files will have a different pattern used. The first overwrite will be a byte of data chosen at random. The second overwrite will be the complement of the first byte. The final overwrite will be a random byte.

The process used to wipe the free space is as follows:

1. A temporary folder (DD_Temp) is created in the root directory of the
   target drive, if one does not exist.

2. Within this folder, a series of files are created and filled with either
   binary zeroes or an alternate data pattern. This process is repeated
   until all the free space on the drive has been overwritten.

3. When finished, the folder and its contents will be removed.

```
************************************
Miscellaneous Patterns
************************************
These patterns cannot be modified.  These processes may be performed from
1 to 99 times as defined by the user.

Loops thru an open file and overwrites each sector with either all
zeroes (Null values (ASCII decimal 0)) or random generated data.
                       or
Internationally-renowned security technologist and author Bruce Schneier
recommends wiping a drive seven times. The first pass overwrites the drive
with Binary 0's, the second with Binary 1's, and the next five with a
randomly generated bit pattern.

    BRUCE  Write 1 - Binary 0's (ASCII decimal 0)
           Write 2 - Binary 1's (ASCII decimal 255)

           Repeat steps 1-2 three times
           Write 7 - A random data stream
           (7 overwrites equal one pass)


If verification is requested then the final pass will be compared with the
data that was supposed to be written.

****************************************
US Government patterns
****************************************
These patterns cannot be modified.  These processes is performed from
1 to 99 times as defined by the user.

Loops thru an open file and overwrites each sector with a pre-defined pattern.

References:
  National Industrial Security Program Operating Manual (NISPOM)
  DoD 5220.22-M, dtd February 28, 2006.
  http://www.dss.mil/isp/fac_clear/download_nispom.html
  Chapter 8 -Information System Security, Section 3-Common Requirements,
  8-301 Clearing and Sanitization

  "Overwrite all addressable locations with a character, its complement, then
  a random character and verify.  THIS METHOD IS NOT APPROVED FOR SANITIZING
  MEDIA THAT CONTAINS TOP SECRET INFORMATION."

  MCSoft MCWipe Functions
  http://www.mcsoft.eu/functions.php?ID=mcwipe&SUB=wipe

  Dban Forum
  http://sourceforge.net/projects/dban/forums/forum/208932

------------------------------------------------------------
U.S. Standard, DoD 5220.22-M (E)  [US Dod Short]
Three overwrites

A standard was developed by the Defence Security Service (DSS) which should
solve the problem of the permanent removal of data for some time. This was
used by many commercial enterprises. Under the National Industrial Security
Program (NISP) representatives of the Industrial Security presented their
security programs. As a part of these NISP the DSS developed the DoD 5220.22-M
standard (National Industrial Security Program Operating Manual - NISPOM).
Which is used in the meantime in almost every deletion tool.

In this manual is beside other procedures the description of a method for the
```

removal of data from magnetic media. The NISP defines a technique for the overwriting of data so that all information is removed from the medium.

```
US DoD Short    Write 1 - One random character
                Write 2 - Complement of previous character
                Write 3 - Random data stream
                (3 overwrites equal one pass)
```

In the regulations of the US American Ministry of Defense people refer expressly to the deletion of information on media with the military classification "Secret" or "TOP-Secret" which is not allowed with this method.

U.S. Standard, DoD 5220.22-M (ECE) [US DoD Long]
Seven overwrites

This method is an extended variant of the DoD 5220.22-M. This variant of the DoD Standard uses for overwriting the data seven runs. Here the data is overwritten two times by using the DoD 5220.22-M (E) standard and one time with random value DoD 5220.22-M (C).

```
Steps 1-3 overwrite the data with the DoD 5220.22-M (E)
          Standard (US DoD Short)
Step 4    overwrite the data with a pseudo random value,
          DoD 5220.22-M (C)
Steps 5-7 overwrite the data with the DoD 5220.22-M (E)
          Standard (US DoD Short) again
```

```
US DoD Long     Write 1 - One random character
                Write 2 - Complement of previous character
                Write 3 - Random data stream
                Write 4 - One random character
                Write 5 - One random character
                Write 6 - Complement of previous character
                Write 7 - Random data stream
                (7 overwrites equal one pass)
```

Verification will be checked to give you a true security overwrite. The final pass will be compared with the data that was supposed to be written.

****
European patterns
****
These patterns cannot be modified. These processes is performed from 1 to 99 times as defined by the user.

Loops thru an open file and overwrites each sector with a pre-defined pattern.

North Atlantic Treaty Organization (NATO) will overwrite the target data area 7 times. For the first 6 passes, each wipe reverses the bit pattern of previous wipe. The 7th overwrite is with a random data stream.

```
    NATO  Write 1 - A random character               (Ex:  ASCII  15  Binary 00001111)
          Write 2 - Complement of previous character (Ex:  ASCII 240  Binary 11110000)

          Repeat steps 1-2 three times
          Write 7 - A random data stream
          (7 overwrites equal one pass)
```

Germany BSI Verschlusssachen-IT-Richtlinien (VSITR) Standard The German Federal Office for IT Security released the VSITR standard, which wipes the data with seven passes. For the first 6 passes, each wipe reverses the bit

pattern of the previous wipe.

Flipping the bits in this way is designed to destabilise the remnants of data that may exist on the edges of the track of the disk to which the data is written. The final pass amplifies this effect by overwriting with 0x55.

German VSITR Write 1 - A random character (Ex: ASCII 15 Binary 00001111)
Write 2 - Complement of previous character (Ex: ASCII 240 Binary 11110000)

Repeat steps 1-2 three times
Write 7 - Data stream of 0x55 (ASCII 85 Binary 01010101)
(7 overwrites equal one pass)

Verification will be checked to give you a true security overwrite. The final pass will be compared with the data that was supposed to be written.

****
Peter Gutmann pattern
****
These patterns cannot be modified. These processes is performed from 1 to 99 times as defined by the user.

Loops thru an open file and overwrites each byte of data with a pre-defined pattern. This process performs 35 writes as described below.

Based on Peter Gutmann`s paper "Secure Deletion of Data from Magnetic and Solid-State Memory", Peter Gutmann 1996, http://www.cs.auckland.ac.nz/~pgut001/secure_del.html

Recommended wiping is done by using the following sequence of 35 consecutive writes to erase data: (overwriting values are given in hex except for random data)

1.	Random	13.	0x33	25.	0xFF
2.	Random	14.	0x44	26.	0x92 0x49 0x24
3.	Random	15.	0x55	27.	0x49 0x24 0x92
4.	Random	16.	0x66	28.	0x24 0x92 0x49
5.	0x55	17.	0x77	29.	0x6D 0xB6 0xDB
6.	0xAA	18.	0x88	30.	0xB6 0xDB 0x6D
7.	0x92 0x49 0x24	19.	0x99	31.	0xDB 0x6D 0xB6
8.	0x49 0x24 0x92	20.	0xAA	32.	Random
9.	0x24 0x92 0x49	21.	0xBB	33.	Random
10.	0x00	22.	0xCC	34.	Random
11.	0x11	23.	0xDD	35.	Random
12.	0x22	24.	0xEE		

(35 overwrites equal one pass)

If verification is requested then the final pass will be compared with the data that was supposed to be written.

==========

Most of the overwrites in the Peter Gutmann wipe are designed to flip bits in MFM/RLL encoded disks, which is an encoding that modern hard disks do not use.

In a followup to his paper, Gutmann said that it is unnecessary to run those passes because you cannot be reasonably certain about how a modern hard disk stores data on the platter. If the encoding is unknown, then writing random patterns is your best

strategy.

In particular, Gutmann says that "in the time since this paper
was published, some people have treated the 35-pass overwrite
technique described in it more as a kind of voodoo incantation to
banish evil spirits than the result of a technical analysis of
drive encoding techniques. As a result, they advocate applying
the voodoo to PRML and EPRML drives even though it will have no
more effect than a simple scrubbing with random data... For any
modern PRML/EPRML drive, a few passes of random scrubbing is the
best you can do".

Read these papers by Peter Gutmann:

    Secure Deletion of Data from Magnetic and Solid-State Memory
    http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html

    Data Remanence in Semiconductor Devices
    http://www.cypherpunks.to/~peter/usenix01.pdf

****
Encrypt pattern
****
There are four encryption algorithms to choose from. These cannot
be modified. These processes is performed one time only.

    Rijndael [US Gov Advanced Encryption Standard (AES)]
    Blowfish [Strong encryption algorithm]
    Twofish [Finalist in NIST encryption contest]
    ArcFour [Strong encryption algorithm]

Loops thru an open file and performs encryption using the appropriate
cipher algorithm. Uses a random generated password 10-50 bytes long.
The key length is also randomly selected based on key lengths available
to the selected algorithm. With Rijndael, the block size is also randomly
selected. The selection process takes place prior to each pass with
each file.

No verification is available since the complete file is processed by
a separate class object.

****
Custom pattern
****
Loops thru an open file and overwrites each sector with a user
defined pattern. The pattern can have up to 5 characters. This
process is performed from 1-99 times as defined by the user. If
verification is requested then the final pass will be compared
with the data that was supposed to be written.

****
** Wipe USB Drive **
****
User selects the USB drive from a drop down box and selects a
pattern from the options window.

Creates a temp folder in the root of the drive to be cleaned. If
one does not exist, a temp folder will be created. Within this
folder, a series of files will be created and filled with data
based on the pattern selected by the user. If this is a flash
drive, the maximum file size will be 10mb. When finished, the
folder and files within are emptied and then deleted.

```
************************************
Random Data
************************************
Random data is generated using a PRNG (Pseudo Random Number
Generator).  I am accessing Microsoft's CryptoAPI (advapi32.dll)
that comes with the Windows operating system. The API
CryptGenRandom() gets its randomness, also known as entropy, from
many sources in Windows.  These include:

    - The current process ID (GetCurrentProcessID).
    - The current thread ID (GetCurrentThreadID).
    - The number of milliseconds since the last boot (GetTickCount).
    - The current time (GetLocalTime).
    - Various high-precision performance counters (QueryPerformanceCounter).
    - A Message Digest-4 (MD4) hash of the user's environment block, which
      includes user name, computer name, and search path. MD4 is a hashing
      algorithm that creates a 128-bit message digest (16 bytes) from input
      data to verify data integrity.
    - High-precision internal CPU counters, such as RDTSC, RDMSR, RDPMC.
    - Low-level system information, such as idle time, kernel time, interrupt
      times, commit limit, page read count, cache read count, nonpaged pool
      allocations, alignment fixup count, operating system look aside
      information.
    - [Optional] User defined data as extra seed data. I created a routine
      named GetExtraSeed() to generate a unique forty byte data string as the
      extra seed data.

Such information is added to a buffer, which is hashed using MD4
and used  as the key to modify a buffer using RC4. (Refer to the
API CryptGenRandom()  the Platform SDK)  The result is a
cryptographic random the user-provided  buffer.)  The result is a
cryptographic random number generator.

References:

Randomize Statement Doesn't Re-initialize Rnd Function
http://support.microsoft.com/default.aspx?scid=kb;en-us;120587

"To re-initialize the random-number generator, use the Rnd
function with a value of -1 and then use the Randomize
statement with the value you want to use as the seed value
for the Rnd function."

VBA's Pseudo Random Number Generator
http://www.noesis.net.au/prng.php

Mark Hutchinson article about the Microsoft Visual BASIC
random number generator.
An Examination of Visual Basic's Random Number Generation
http://www.15seconds.com/issue/051110.htm

INFO: How Visual Basic Generates Pseudo-Random Numbers for
the RND Function
http://support.microsoft.com/kb/231847/en-us

RND and RANDOMIZE Alternatives for Generating Random Numbers
http://support.microsoft.com/kb/28150/EN-US/
```

```
================================================================
                Sample Log file Data
================================================================

DiscDataWipe Log File
dd.mm.yyyy hh:mm:ss  Description
********************************************************************************
24.07.2010 06:21:33  STARTED: 1 Pass using Rijndael (AES) Encryption  [High Security]
24.07.2010 06:21:33         Rijndael parms:  Pwd len-36  Key len-224  Block size-224
24.07.2010 06:21:34     File:  C:\Temp\Testresults\Mt11231b\MTB_DH.TXT (58,129 bytes)
24.07.2010 06:21:34         Rijndael parms:  Pwd len-14  Key len-128  Block size-160
24.07.2010 06:21:35     File:  C:\Temp\Testresults\Mt11231b\MTB_Ent.txt (27,681 bytes)
24.07.2010 06:21:35         Rijndael parms:  Pwd len-31  Key len-192  Block size-160
24.07.2010 06:21:35     File:  C:\Temp\Testresults\Mt11231b\MTB_Test01.txt (19,739
bytes)
24.07.2010 06:21:35         Rijndael parms:  Pwd len-36  Key len-192  Block size-256
24.07.2010 06:21:36     File:  C:\Temp\Testresults\Mt11231b\MTB_Test02.txt (13,605
bytes)
24.07.2010 06:21:36         Rijndael parms:  Pwd len-25  Key len-192  Block size-160
24.07.2010 06:21:36     File:  C:\Temp\Testresults\Mt11231b\MTB_Test03.txt (1,052
bytes)
24.07.2010 06:21:36         Rijndael parms:  Pwd len-40  Key len-192  Block size-192
24.07.2010 06:21:36     File:  C:\Temp\Testresults\Mt11231b\MTB_Test04.txt (293 bytes)
24.07.2010 06:21:46     Folder:  C:\Temp\Testresults\Mt11231b  [DELETED]
24.07.2010 06:21:47  FINISHED:  1 folders, 6 files (120,499 bytes (117.7 KB))
24.07.2010 06:23:37  STARTED: 1 Pass using Peter Gutmann  [ 35 writes ]  [High
Security]
24.07.2010 06:23:49     File:  C:\Temp\Testresults\Mt19937\MT_DH.TXT (58,129 bytes)
24.07.2010 06:23:53     File:  C:\Temp\Testresults\Mt19937\MT_Ent.txt (27,680 bytes)
24.07.2010 06:23:56     File:  C:\Temp\Testresults\Mt19937\MT_Test01.txt (19,738
bytes)
24.07.2010 06:23:59     File:  C:\Temp\Testresults\Mt19937\MT_Test02.txt (13,604
bytes)
24.07.2010 06:24:00     File:  C:\Temp\Testresults\Mt19937\MT_Test03.txt (1,051 bytes)
24.07.2010 06:24:01     File:  C:\Temp\Testresults\Mt19937\MT_Test04.txt (292 bytes)
24.07.2010 06:24:11     Folder:  C:\Temp\Testresults\Mt19937  [DELETED]
24.07.2010 06:24:12  FINISHED:  1 folders, 6 files (120,494 bytes (117.7 KB))
24.07.2010 06:24:35  STARTED: 1 Pass using Bruce Schneier  [ 7 writes ]  [High
Security]
24.07.2010 06:24:36     File:  C:\Temp\TestResults\TT800\TT8_DH.TXT (58,129 bytes)
24.07.2010 06:24:36     File:  C:\Temp\TestResults\TT800\TT8_Ent.txt (27,688 bytes)
24.07.2010 06:24:37     File:  C:\Temp\TestResults\TT800\TT8_Test01.txt (19,721 bytes)
24.07.2010 06:24:37     File:  C:\Temp\TestResults\TT800\TT8_Test02.txt (13,587 bytes)
24.07.2010 06:24:38     File:  C:\Temp\TestResults\TT800\TT8_Test03.txt (1,034 bytes)
24.07.2010 06:24:38     File:  C:\Temp\TestResults\TT800\TT8_Test04.txt (275 bytes)
24.07.2010 06:24:47     Folder:  C:\Temp\TestResults\TT800  [DELETED]
24.07.2010 06:24:54     Folder:  C:\Temp\TestResults  [DELETED]
24.07.2010 06:24:55  FINISHED:  2 folders, 6 files (120,434 bytes (117.6 KB))
24.07.2010 06:27:31  STARTED: Wipe free space on drive C:\
24.07.2010 06:27:31     Creating files filled with binary 0's on drive C:\
24.07.2010 06:27:43     Finished creating 9 temp files on drive C:\
24.07.2010 06:27:43     Using pattern Binary 0's  [ Ignore space messages ]
24.07.2010 06:27:51     Removed 9 temp files from drive C:\
24.07.2010 06:27:51     Pass 2 of 3 using pattern Binary 0's  [ Ignore space messages
]
24.07.2010 06:27:51     Creating files filled with binary 0's on drive C:\
24.07.2010 06:27:52     Finished creating 9 temp files on drive C:\
24.07.2010 06:27:59     Removed 9 temp files from drive C:\
24.07.2010 06:27:59     Pass 3 of 3 using pattern Binary 0's  [ Ignore space messages
]
24.07.2010 06:27:59     Creating files filled with binary 0's on drive C:\
24.07.2010 06:27:59     Finished creating 9 temp files on drive C:\
```

24.07.2010 06:28:06   Removed 9 temp files from drive C:\
24.07.2010 06:28:06 FINISHED: Wipe free space on drive C:\ Amount overwritten:
35,433,476,096 bytes (33.0 GB)

==========

License

Kenneth Ives
kenaso@tx.rr.com

==========

Preamble

This License governs Your use of the Work. This License is intended to allow developers to use the Source Code and Executable Files provided as part of the Work in any application in any form.

The main points subject to the terms of the License are:

- Source Code and Executable Files can be used in commercial applications;
- Source Code and Executable Files can be redistributed; and
- Source Code can be modified to create derivative works.

No claim of suitability, guarantee, or any warranty whatsoever is provided. The software is provided "as-is".

This License is entered between You, the individual or other entity reading or otherwise making use of the Work licensed pursuant to this License and the individual or other entity which offers the Work under the terms of this License ("Author").

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CODE PROJECT OPEN LICENSE ("LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HEREIN, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE AUTHOR GRANTS YOU THE RIGHTS CONTAINED HEREIN IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO ACCEPT AND BE BOUND BY THE TERMS OF THIS LICENSE, YOU CANNOT MAKE ANY USE OF THE WORK.

Definitions.

"Articles" means, collectively, all articles written by Author which describes how the Source Code and Executable Files for the Work may be used by a user.

"Author" means the individual or entity that offers the Work under the terms of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works.

"Executable Files" refer to the executables, binary files, configuration and any required data files included in the Work.

"Publisher" means the provider of the website, magazine, CD-ROM, DVD or other medium from or by which the Work is obtained by You.

"Source Code" refers to the collection of source code and configuration files used to create the Executable Files.

"Standard Version" refers to such a Work if it has not been modified, or has been modified in accordance with the consent of the Author, such consent being in the full discretion of the Author.

"Work" refers to the collection of files distributed by the Publisher, including the Source Code, Executable Files, binaries, data files, documentation, whitepapers and the Articles.

"You" is you, an individual or entity wishing to use the Work and exercise your rights under this License.

Fair Use/Fair Use Rights. Nothing in this License is intended to reduce, limit, or restrict any rights arising from fair use, fair dealing, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

License Grant. Subject to the terms and conditions of this License, the Author hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

You may use the standard version of the Source Code or Executable Files in Your own applications.

You may apply bug fixes, portability fixes and other modifications obtained from the Public Domain or from the Author. A Work modified in such a way shall still be considered the standard version and will be subject to this License.

You may otherwise modify Your copy of this Work (excluding the Articles) in any way to create a Derivative Work, provided that You insert a prominent notice in each changed file stating how, when and where You changed that file.

You may distribute the standard version of the Executable Files and Source Code or Derivative Work in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution.

The Articles discussing the Work published in any form by the author may not be distributed or republished without the Author's consent. The author retains copyright to any such Articles. You may use the Executable Files and Source Code pursuant to this License but you may not repost or republish or otherwise distribute or make available the Articles, without the prior written consent of the Author.

Any subroutines or modules supplied by You and linked into the Source Code or Executable Files this Work shall not be considered part of this Work and will not be subject to the terms of this License.

Patent License. Subject to the terms and conditions of this License, each Author hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, import, and otherwise transfer the Work.

Restrictions. The license granted in Section 3 above is expressly

made subject to and limited by the following restrictions:

You agree not to remove any of the original copyright, patent, trademark, and attribution notices and associated disclaimers that may appear in the Source Code or Executable Files.

You agree not to advertise or in any way imply that this Work is a product of Your own.

The name of the Author may not be used to endorse or promote products derived from the Work without the prior written consent of the Author.

You agree not to sell, lease, or rent any part of the Work.

You may distribute the Executable Files and Source Code only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy of the Executable Files or Source Code You distribute and ensure that anyone receiving such Executable Files and Source Code agrees that the terms of this License apply to such Executable Files and/or Source Code. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute the Executable Files or Source Code with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License.

You agree not to use the Work for illegal, immoral or improper purposes, or on pages containing illegal, immoral or improper material. The Work is subject to applicable export laws. You agree to comply with all such laws and regulations that may apply to the Work after Your receipt of the Work.

Representations, Warranties and Disclaimer. THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

Indemnity. You agree to defend, indemnify and hold harmless the Author and the Publisher from and against any claims, suits, losses, damages, liabilities, costs, and expenses (including reasonable legal or attorneys' fees) resulting from or relating to any use of the Work by You.

Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL THE AUTHOR OR THE PUBLISHER BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK OR OTHERWISE, EVEN IF THE AUTHOR OR THE PUBLISHER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Termination.

This License and the rights granted hereunder will terminate automatically upon any breach by You of any term of this License. Individualss or entities who have received Derivative Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 6, 7, 8, 9, 10 and 11 will survive any termination of this License.

If You bring a copyright, trademark, patent or any other infringement claim against any contributor over infringements You claim are made by the Work, your License from such contributor to the Work ends automatically.

Subject to the above terms and conditions, this License is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, the Author reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

Publisher. The parties hereby confirm that the Publisher shall not, under any circumstances, be responsible for and shall not have any liability in respect of the subject matter of this License. The Publisher makes no warranty whatsoever in connection with the Work and shall not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. The Publisher reserves the right to cease making the Work available to You at any time without notice

Miscellaneous.

This License shall be governed by the laws of the location of the head office of the Author or if the Author is an individual, the laws of location of the principal place of residence of the Author.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this License, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed herein. There are no understandings, agreements or representations with respect to the Work not specified herein. The Author shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Author and You.

Kenneth Ives kenaso@tx.rr.com