kiWipe.dll (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.

Important note:
    If you have less than five (5) MB of freespace left on your hard drive,
    you need to be doing some major clean up. You should be either removing
    obsolete files and folders or upgrading to a larger capacity disk.

=
*** 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 the implementation of the secure hash algorithms
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and
SHA-512/256, can be found 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

Export Control: Certain cryptographic devices and technical
data regarding them are subject to Federal export controls.
Exports of cryptographic modules implementing this standard
and technical data regarding them must comply with these
Federal regulations and be licensed by the Bureau of Export
Administration of the U.S. Department of Commerce.
Information about export regulations is available at:
http://www.bis.doc.gov/index.htm

****

****
** 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

```
================================================================================
Available in cWipe (clsWipe)
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  Public Enum enumWipeMethod
      eWipe_AlternateMethod   '  0 - Use alternate method to wipe freespace
      eWipe_BinaryZeroes      '  1 - Binary zeroes (0x00)
      eWipe_RandomData        '  2 - Random genereated data
      eWipe_US_DoD_Short      '  3 - US DoD 5220.22-M E
      eWipe_US_DoD_Long       '  4 - US DoD 5220.22-M ECE
      eWipe_NATO              '  5 - North Atlantic Treaty Organization (NATO)
      eWipe_GermanVISTR       '  6 - Germany BSI Verschlusssachen-IT-Richtlinien
(VSITR)
      eWipe_BruceSchneier     '  7 - Bruce Schneier (security technologist,
cryptoanalyst)
      eWipe_PeterGutmann      '  8 - Peter Gutmann (Computer Science professor)
      eWipe_Rijndael          '  9 - Encryption [Rijndael - NIST winner] AES [Advanced
Encryption Standard]
      eWipe_Blowfish          ' 10 - Encryption [Blowfish - Strong, reliable and fast]
      eWipe_Twofish           ' 11 - Encryption [Twofish  - NIST runner up]
      eWipe_ArcFour           ' 12 - Encryption [ArcFour  - Strong, reliable and fast]
      eWipe_CustomPattern     ' 13 - User defined pattern
  End Enum

  Public Enum enumEncryptAlgo
      eAlgo_Rijndael  ' 0 - Rijndael algorithm (AES)
      eAlgo_Blowfish  ' 1 - Blowfish algorithm
      eAlgo_Twofish   ' 2 - Twofish algorithm
      eAlgo_ArcFour   ' 3 - ArcFour algorithm
  End Enum


' ************************************************************************
' ****                        Properties                              ****
' ************************************************************************
  AlternateMethod - Input/Output - Boolean data to designate if the process is to use
US DoD Short
             pattern to wipe the free space of a drive.
             Syntax:  X.AlternateMethod = TRUE        (Input)
                      Debug.Print X.AlternateMethod   (Output)

  CurrentPattern - Input only - String data designating the current pattern in use.
             Syntax:  X.CurrentPattern = "Zero fill sectors [ 0x00 ]"

  DeleteTopLevelFolder - Input only - Boolean data to designate if the top level
folder is to be
             deleted when finished processing folders.  This flag is ignored if
DoFolders = False.
             Syntax:  X.DeleteTopLevelFolder = TRUE

  DisplayMsgs - Input only - Boolean data to designate if any exception messages
should
             be displayed on the screen during the verification process. If VerifyData
is
             FALSE then this switch is ignored.
             Syntax:  X.DisplayMsgs = FALSE

  DoFolders - Input/Output - Boolean data to designate if the process is to wipe all
the contents
             of a folder.
```

```
            Syntax:  X.DoFolders = TRUE        (Input)
                     Debug.Print X.DoFolders   (Output)

  DoSubFolders - Input only - Boolean data to designate if the wipe process is to
include any
            subfolders.  This flag is ignored if DoFolders = False.
            Syntax:  X.DoSubFolders = TRUE

  EncryptAlgo - Input only - Long integer designating which encryption algorithm to
use.
            Syntax: X.EncryptAlgo = eArcFour

  FlashDrive - Input/Output - Boolean data to designate if processing a Flash drive.
This is
            ignored if USB_Drive is FALSE.
            Syntax:  X.FlashDrive = TRUE        (Input)
                     Debug.Print X.FlashDrive   (Output)

  FloppyDrive - Input/Output - Boolean data to designate if the target area is a
floppy disk.
            Syntax:  X.FloppyDrive = TRUE        (Input)
                     Debug.Print X.FloppyDrive   (Output)

  KeepFolderStructure - Input only - Boolean data to designate if the wipe process is
to remove
            files only and leave the directory structure in tact.
            Syntax:  X.KeepFolderStructure = TRUE

  LogData  - Input only - Boolean data to designate if the log file is to be updated.
            Syntax:  X.LogData = True

  LogPattern - Input only - String data to be used to describe what pattern is being
            currently used.
            Syntax:  X.LogPattern = "5 writes per sector [ (Random * 4), 0x00 ]"

  Passes - Input only - Long integer data to designate the number of passes to be
performed.
            Syntax:  X.Passes = 1

  StopProcessing - Input/Output - Boolean data to designate if the user has opted to
stop the
            processing.
            Syntax:  X.StopPressed = TRUE        (Input)
                     Debug.Print X.StopPressed   (Output)

  TypeTarget - Input/Output - Long integer data to designate if files or folders are
to be processed.
            1 = Files identified by the user
            2 = Folders
            3 = USB drive

            Syntax:  X.TypeTarget = 1           (Input)
                     Debug.Print X.TypeTarget   (Output)

  USB_Drive - Input/Output - Boolean data to designate if processing a USB drive
            Syntax:  X.USB_Drive = TRUE        (Input)
                     Debug.Print X.USB_Drive   (Output)

  VerifyData - Input only - Boolean data to designate if the last pass is to be
verified.

  Version - Output - String - Name of DLL and version information
```

```
  WipeFreeSpace - Input/Output - Boolean data to designate if the process selected is
to wipe
              the free space of a drive.
              Syntax:  X.WipeFreeSpace = TRUE        (Input)
                       Debug.Print X.WipeFreeSpace   (Output)

  WipeMethod - Input/Output - Long integer data to designate the wiping pattern to be
used.
              Syntax:  X.WipeMethod = WIPE_DOD_LONG  (Input)
                       Debug.Print X.WipeMethod      (Output)

  WipePath - Input only - String data to be displayed as to what path is being
accessed.
              Syntax:  X.WipePath = "Wipe free space on drive C:"

  WipePatterns - Input only - Variant array of data to designate the wiping pattern to
be used
              for Custom wipes.
              Syntax:  X.WipePatterns = array of data

  ZeroLastWrite - Input Only - Boolean flag to designate if the last write should be
all zeroes
                   (null values)  Only valid with options 2-7.


' ************************************************************************
' ****                          Public Events                         ****
' ************************************************************************
  ' maintain an accurate count of the passes performed
  Public Event CountPasses(ByVal lngCurrentPass As Long, _
                           ByVal lngMaxPasses As Long)

  ' Used to display the accumulation count as it is incremented.
  Public Event CountTotals(ByVal strPathFile As String, _
                           ByVal dblFolderCount As Double, _
                           ByVal dblFileCount As Double, _
                           ByVal dblByteCount As Double, _
                           ByVal dblFileSize As Double, _
                           ByVal blnCountOnly As Boolean)

  ' Raises the name of the current pattern being used.
  Public Event CurrentPattern(ByVal strCurrentPattern As String)

  ' maintain an accurate count of the bytes written
  Public Event CurrentProgress(ByVal dblByteCount As Double, _
                               ByVal dblMaxAmount As Double)

  ' maintain an accurate count of the overall progress
  Public Event OverallProgress(ByVal dblByteCount As Double, _
                               ByVal dblMaxAmount As Double)

  ' Current position on a progressbar
  Public Event FileProgress(ByVal lngFileProgress As Long)
  Public Event TotalProgress(ByVal lngTotalProgress As Long)

  ' Display Elapsed time, time remaining and transfer rate
  Public Event ElapsedTime(ByVal strElapsedTime As String)

  Public Event TimeRemaining(ByVal strTimeRemaining As String, _
                             ByVal strTransferRate As String)

  ' Display a Stand-by message
  Public Event WaitMsg(ByVal strMsg As String)
```

```
' ************************************************************************
' ****                          Methods                                 ****
' ************************************************************************
' This is where the array of file names or the name of the folder is passed
' inside an array to be wiped.  Also called if wiping a USB drive.
'
'  Syntax of the array:
'
'     One file selected:
'       Array(0) = "C:\foldername\File1.dat"
'
'     More than one file selected:
'       Array(0) = "C:\foldername"   ' Name of folder
'       Array(1) = "File1.dat"
'       Array(2) = "File2.dat"
'
'     Folder to be processed:
'       Array(0) = "C:\foldername"
BeginProcessing(ByVal vntFileList As Variant) As Boolean

' Called to wipe a drive's free space
WipeTheFreeSpace(ByVal strDrive As String) As Boolean
```

```
================================================================================
Available in cPRNG (clsRandom)
A cryptographically random number generator using Microsoft's CryptoAPI.
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  Public Enum enumPRNG_ReturnFormat
      ePRNG_ASCII           ' 0
      ePRNG_HEX             ' 1
      ePRNG_HEX_ARRAY       ' 2
      ePRNG_BYTE_ARRAY      ' 3
      ePRNG_LONG_ARRAY      ' 4
      ePRNG_DBL_ARRAY       ' 5
  End Enum

  Public Enum enumPRNG_HashAlgorithm
      ePRNG_MD2             ' 0
      ePRNG_MD4             ' 1
      ePRNG_MD5             ' 2
      ePRNG_SHA1            ' 3
      ePRNG_SHA256          ' 4
      ePRNG_SHA384          ' 5
      ePRNG_SHA512          ' 6
  End Enum

  Public Enum enumPRNG_Compare
      ePRNG_CaseSensitive   ' 0 - Exact byte match
      ePRNG_IgnoreCase      ' 1 - Uppercase/Lowercase considered same
  End Enum

' ************************************************************************
' ****                       Properties                               ****
' ************************************************************************

  StopProcessing - Input/Output - Boolean - True if user wants to stop processing

  AES_Ready - Output - Boolean - True if operating system can use SHA2 functionality

  CompareMethod - Input - Long Integer - Designates type of data comparison to be used

' ************************************************************************
' ****                       Methods                                  ****
' ************************************************************************
' Build random data using ASCII values 0-255.
Function BuildRndData(ByVal lngDataLength As Long, _
             Optional ByVal lngReturnFormat As enumPRNG_ReturnFormat =
ePRNG_BYTE_ARRAY, _
             Optional ByVal blnCreateExtraSeed As Boolean = True) As Variant

' Build random data that falls between two ASCII values, inclusive.
Function BuildWithinRange(ByVal lngDataLength As Long, _
                 Optional ByVal lngLowValue As Long = 0, _
                 Optional ByVal lngHighValue As Long = 255, _
                 Optional ByVal lngRetDataType As enumPRNG_ReturnFormat =
enuByteArray, _
                 Optional ByVal blnCreateExtraSeed As Boolean = True) As Variant

' The data will be SORTED.  This routine removes all duplicates based on
' user selection of case sensitivity. The number of duplicates removed
' are returned.
Function RemoveDupes(ByRef avntData As Variant, _
```

```
            Optional ByRef lngDupeCnt As Long = 0, _
            Optional ByVal blnReturnMixed As Boolean = False) As Boolean

' An array of data passed to this routine will be rearranged.
Sub ReshuffleData(ByRef avntData As Variant, _
         Optional ByVal lngMixCount As Long = 25)

' With this routine you can generate a series of non-repeating numbers.
' An array will be loaded starting with the base number (lngMinValue)
' requested up to the maximum value requested (lngMaxValue).  You can
' also enter the incremental step between the minimum and maximum value.
' This array is then passed to another routine ReshuffleData() to be
' throughly rearranged.  When it is returned, the requested number of
' elements (lngReturnQty) from the mixed array are transferred
' sequentially to the return array (alngMixed()).
'
' Syntax:  x = NonRepeatingNbrs(100, 0, 9999, 5)
'          Return 100 numbers, lowest = 0, highest = 9999,
'          incremental step = 5, Sort return data in
'          Ascending order (default)
Function NonRepeatingNbrs(ByVal lngReturnQty As Long, _
                          ByVal lngMinValue As Long, _
                          ByVal lngMaxValue As Long, _
                 Optional ByVal lngStep As Long = 1, _
                 Optional ByVal blnSortData As Boolean = True) As Long()

' CombSort is faster than all but QuickSort and close to it.  On the
' other hand, the code is much simpler than QuickSort and can be easily
' customized for any array type.  The CombSort was first published by
' Richard Box and Stephen Lacey in the April 1991 issue of Byte magazine.
Function CombSort(ByRef avntData As Variant, _
         Optional ByVal blnAscending As Boolean = True) As Boolean

' Generate a one-way hash string from a string of data. These are the
' algorithms to use:   MD2  MD4  MD5  SHA-1  SHA-256  SHA-384  SHA-512
'
' Special note:  SHA-224, SHA-512/224 and SHA-512/256 have not yet been
' implemented into the Microsoft crypto suite of hashes.
Function CreateHash(ByVal strInput As String, _
           Optional ByVal lngHashAlgo As enumPRNG_HashAlgorithm = ePRNG_SHA512, _
           Optional ByVal blnReturnAsHex As Boolean = True) As String

' Generate a random long integer between two input values.
Function GetRndValue(ByVal sngLow As Single, _
                     ByVal sngHigh As Single) As Long

' Convert a long integer to a double precision number.  Returns a decimal
' position of 14 places.
Function LongToDouble(ByVal lngValue As Long) As Double

' This is an ArrPtr function that determines if the passed array is
' initialized, and if so will return the pointer to the safearray header.
' If the array is not initialized, it will return zero.
' Syntax:   If CBool(IsArrayInitialized(array_being_tested)) Then ...
Function IsArrayInitialized(ByVal avntData As Variant) As Long

' Properly empty and deactivate a collection
Sub EmptyCollection(ByRef colData As Collection)

' This little code snippet returns a truly random value.
Function RndSeed() As Double

' Swap data with each other.  Wrote this function since BASIC stopped
```

```
' having its own SWAP function.  Use this for swapping strings, type
' structures, numbers with decimal values, etc.
Sub SwapData(ByRef vntData1 As Variant, _
             ByRef vntData2 As Variant)

' Swap numeric data (byte, integer, or long) with each other
' without using a temporary holding variable.
Sub SwapLong(ByRef AA As Long, _
             ByRef BB As Long)

Sub SwapInt(ByRef AA As Integer, _
            ByRef BB As Integer)

Sub SwapByte(ByRef AA As Byte, _
             ByRef BB As Byte)

' Converts a byte array to string data.
Function ByteArrayToString(ByRef abytData() As Byte) As String

' Converts string data to a byte array.
Function StringToByteArray(ByVal strData As String) As Byte()

' Creates a unique string of hex data using CryptoAPI hash functions.  Also,
' randomly select a starting position in hashed data string to capture two
' eight byte strings of data.  These will be converted into long integers
' for new carryover values.
Function CreateExtraSeed(Optional ByVal lngRetLength As Long = 0) As String
```

```
================================================================================
Available in cDiskInfo (clsDiskInfo)
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  Public Enum enumDiskSpace
      eFreespace         ' 0
      eTotalSize         ' 1
      eUsedSpace         ' 2
      eAvailSpace        ' 3
  End Enum

  Public Enum enumDriveType
      eUnknown           ' 0 Unknown drive type
      eBadRoot           ' 1 No root directory
      eRemovable         ' 2 Floppy or Jaz drive
      eFixed             ' 3 Local hard drive
      eNetwork           ' 4 Shared Network drive
      eCDRom             ' 5 CD-Rom drive (CD or DVD)
      eRamdisk           ' 6 Virtual memory disk
  End Enum

  Public Enum enumIDE_DRIVE_NUMBER
      ePrimaryMaster     ' 0
      ePrimarySlave      ' 1
      eSecondaryMaster   ' 2
      eSecondarySlave    ' 3
      eTertiaryMaster    ' 4
      eTertiarySlave     ' 5
      eQuartiaryMaster   ' 6
      eQuartiarySlave    ' 7
  End Enum

  ' Status Flags Values
  Public Enum enumSTATUS_FLAGS
      ePrefailureWarranty = &H1
      eOnLineCollection = &H2
      ePerformanceAttribute = &H4
      eErrorRateAttribute = &H8
      eEventCouontAttribute = &H10
      eSELF_PRESERVING_ATTRIBUTE = &H20
  End Enum


' ************************************************************************
' ****                        Properties                              ****
' ************************************************************************
  StopProcessing - Input/Output - Boolean - True if user wants to stop processing

  DriveType - Output - String - Type of drive (Ex:  Physical Hard Drive)

  DriveTypeExtra - Output - String - Type of drive extras (Ex:  Fixed Hard Drive)

  FormattedSize - Output - String - Formatted drive size (Ex:  70.1 GB)

  Partition - Output - String - Partition information (Ex:  Disk #0, Partition #1)

  VolumeName - Output - String - Name of partition.  Changes with each format.
                                 (Ex:  HP_PAVILION)

  VolumeSerial - Output - String - Serial number assigned to partition after each
```

format. (Ex: 4FC0-112D)

FileSysType - Output - String - Type file system assigned during format (Ex: NTFS)

MfgHDSerial - Output - String - Manufacturer hard drive serial number. Cannot be changed by user.

MfgHDModel - Output - String - Manufacturer hard drive model number. Cannot be changed by user.

MfgHDFirmware - Output - String - Manufacturer hard drive firmware number. Cannot be changed by user.

BytesPerSector - Output - Long - Bytes per sector as defined when disk is formatted.

BytesPerCluster - Output - Long - Bytes per sector as defined when disk is formatted.

SectorsPerTrack - Output - Long - Sectors assigned to each track.

TotalCylinders - Output - Currency - Total number of cylinders per hard drive.

TracksPerCylinder - Output - Long - Number of tracks per cylinder.

TotalSectors - Output - Long - Total number of sectors per hard drive.

TotalClusters - Output - Long - Total number of clusters per hard drive.

FreeClusters - Output - Long - Total number of free clusters per hard drive.

TotalBytesFree - Output - Currency - Total number of free bytes per hard drive.

AvailableBytes - Output - Currency - Total number of available bytes per hard drive.

UsedSpace - Output - Currency - Amount of used space on hard drive.

TotalDiskSpace - Output - Currency - Total amount of space on hard drive.

PartitionSpace - Output - Currency - Amount of space on hard drive after formatting.

```
' ************************************************************************
' ****                        Methods                                 ****
' ************************************************************************

' Captures information about a specific drive into a type data structure.
Sub GetDriveInfo(ByVal strDrive As String)

' Make an API call to a specific drive and capture the volume information.
' Returns a brief desription of drive (ex: "Remote (network) drive").
Function GetDriveDescription(ByVal strDrive As String) As String

' Determines if the drive is a CD-Rom drive and returns its handle.
Function IsCDRomDrive(ByVal strDrive As String) As Long

' Make an API call to a specific drive and capture the volume information
Function GetClusterSize(ByVal strDrive As String) As Long

' Capture a list of all available drive letters  (Ex: A:\, C:\, D:\, etc.)
Function GetDriveLetters() As String()

' Make an API call to a specific drive and capture the volume information
Sub GetVolumeInfo(ByVal strDrive As String, _
```

```
         Optional ByRef strVolName As String = "", _
         Optional ByRef strVolSerialNo As String = "", _
         Optional ByRef strFileSysType As String = "")

' Used to determine if a drive is of a specific type.  True - if there is
' a match else FALSE is returned.
Function SpecificTypeOfDrive(ByVal strDrive As String, _
                             ByVal lngTypeNeeded As enumDriveType) As Boolean

' Capture the specific space information about a selected drive.
Function GetDiskSpaceInfo(ByVal strDrive As String, _
                 Optional ByVal lngChoice As enumDiskSpace = eFreeSpace) As Currency

' Determine if a device is functioning.
Function IsDeviceReady(ByVal strDrive As String) As Boolean

' Lock a device.  True - Lock the device.  FALSE - Unlock the device.
Sub DeviceLock(ByVal strDrive As String, _
               ByVal blnLockDevice As Boolean)

' Use WMI (Windows Management Instrumentation) to obtain a drive's partition
' information and the disk number as assigned by the BIOS.
Function GetDiskNumber(ByVal strDrive As String) As Long

' Get hard disk manufacturer information.  Default is the first physical drive.
' EX:  strModel = "QUANTUM FIREBALLP AS20.5"
Sub GetMfrInfo(Optional ByVal lngDrvNumber As enumIDE_DRIVE_NUMBER = ePrimaryMaster, _
               Optional ByRef strSerial As String = "", _
               Optional ByRef strModel As String = "", _
               Optional ByRef strFirmware As String = "")

' Return a string representing the value in string format to requested number
' of decimal positions.   (Ex:  2,530,096 bytes --> 2.4 MB)
Function DisplayNumber(ByVal dblCapacity As Double, _
                       Optional ByVal lngDecimals As Long = 1) As String

' Create a nested directory structure.
' EX:  strPath = "C:\Program Files\MyDir\Sub_1\Sub_2"
Function CreateDirStructure(ByVal strPath As String) As Boolean

' See if CD drive is physically mounted or just a drive letter (Virtual)
Function IsCDMounted(ByVal strDrive As String) As Boolean
```

```
================================================================================
Available in cOperSystem (clsOperSystem)
================================================================================

' ************************************************************************
' ****                       Properties                                 ****
' ************************************************************************
  VersionName - Output - String data - the operating system version name.
                Example: Windows 2000

  VersionNumber - Output - String data - the operating system version number.
                  Example: 5.00

  BuildNumber - Output - String data - the operating system build number.
                Example: 2195

  VersionData - Output - String data - the full operating system version name.
                Example: Windows 2000 5.00.2195 Service Pack 4

  ServicePack - Output - String data - In Win9x, this can be any arbitrary
                string provided by the manufacturer. In NT based operating systems,
                this is the service pack.

  WinPlatformID -  Output - Long integer - Represents the operating system platform ID

  bWindowsNT - Output - Boolean - True if operating system is Windows NT based

  bWinNT4orNewer - Output - Boolean - True if operating system is Windows NT4 or newer

  bWin2000orNewer - Output - Boolean - True if operating system is Windows 2000 or
newer

  bWinXPorNewer - Output - Boolean - True if operating system is Windows Xp or newer

  bWinVistaOrNewer - Output - Boolean - True if operating system is Windows Vista or
newer

  bWin2000 - Output - Boolean - True if operating system is Windows 2000

  bWin2000Pro - Output - Boolean - True if operating system is Windows 2000
Professional

  bWin2000Workstation - Output - Boolean - True if operating system is Windows 2000
Workstation

  bWin2000Server - Output - Boolean - True if operating system is Windows 2000 Server

  bWin2000DatacenterSvr - Output - Boolean - True if operating system is Windows 2000
Datacenter Server

  bWin2000AdvancedSvr - Output - Boolean - True if operating system is Windows 2000
Advanced Server

  bWinXP - Output - Boolean - True if operating system is Windows XP

  bWinXPSP2 - Output - Boolean - True if operating system is Windows XP with SP2

  bWinXPHomeEdition - Output - Boolean - True if operating system is Windows XP Home
Edition

  bWinXPPro - Output - Boolean - True if operating system is Windows XP Professional
```

bWinXPMediaCenter - Output - Boolean - True if operating system is Windows XP Media Center

bWinXPStarter - Output - Boolean - True if operating system is Windows XP Starter OS

bWinXPTabletPC - Output - Boolean - True if operating system is Windows XP Tablet PC

bWinXPEmbedded - Output - Boolean - True if operating system is Windows XP Embedded

bWinVista - Output - Boolean - True if operating system is Windows Vista

bWinVistaSP1 - Output - Boolean - True if operating system is Windows Vista with Service Pack 1

bWinVistaHomeBasic - Output - Boolean - True if operating system is Windows Vista Home Basic

bWinVistaHomeEdition - Output - Boolean - True if operating system is Windows Vista Home Edition

bWinVistaHomePremium - Output - Boolean - True if operating system is Windows Vista Home Premium

bWinVistaHomeServer - Output - Boolean - True if operating system is Windows Vista Home Server

bWinVistaUltimate - Output - Boolean - True if operating system is Windows Vista Ultimate

bWinVistaBusiness - Output - Boolean - True if operating system is Windows Vista Business

bWinVistaEnterprise - Output - Boolean - True if operating system is Windows Vista Enterprise

bWinVistaWorkstation - Output - Boolean - True if operating system is Windows Vista Workstation

bWinVistaStarter - Output - Boolean - True if operating system is Windows Vista Starter edition

bWindows7 - Output - Boolean - True if operating system is Windows 7

bWin2003 - Output - Boolean - True if operating system is Windows 2003

bWin2003Server - Output - Boolean - True if operating system is Windows 2003 Server

bWin2003ServerR2 - Output - Boolean - True if operating system is Windows 2003 Server Rel 2

bWin2003StorageServer - Output - Boolean - True if operating system is Windows 2003 Storage Server

bBladeServer - Output - Boolean - True if this PC is a Blade Server

bWebServer - Output - Boolean - True if this PC is a Web Server

bWinHomeServer - Output - Boolean - True if this PC is a Home Server

bClusterServer - Output - Boolean - True if this PC is a Cluster Server

bComputeClusterServer - Output - Boolean - True if this PC is a Compute Cluster Server

bWinServer2008 - Output - Boolean - True if this PC is a Windows Server 2008

bWinServer2008R2 - Output - Boolean - True if this PC is a Windows Server 2008 R2

bDataCenterServer - Output - Boolean - True if operating system is Windows XP Datacenter

bDataCenterServerCore - Output - Boolean - True if operating system is Windows XP Datacenter Core

bBackOfficeServer - Output - Boolean - True if this PC is a Backoffice Server

bDomainController - Output - Boolean - True if this PC is a Domain Controller

bEnterpriseServer - Output - Boolean - True if this PC is a Enterprise Server

bEnterpriseServerCore - Output - Boolean - True if this PC is a Enterprise Server Core

bTerminalServer - Output - Boolean - True if this PC is a Terminal Server

bSmallBusinessServer - Output - Boolean - True if this PC is a Small Business Server

bSmallBusinessServerPremium - Output - Boolean - True if this PC is a Small Business Server Premium

bSmallBusinessRestrictedServer - Output - Boolean - True if this PC is a Small Business Restricted Server

bStandardServer - Output - Boolean - True if this PC is a Standard Server

bStandardServerCore - Output - Boolean - True if this PC is a Standard Server Core

bOperSystem64 - Output - Boolean - True if operating system is Windows 64-bit

bWinVista64 - Output - Boolean - True if operating system is Windows Vista 64-bit

bWinXPPro64 - Output - Boolean - True if operating system is Windows XP 64-bit Professional

bDatacenterItanium64 - Output - Boolean - True if operating system is Windows XP 64-bit Datacenter Itanium

bEnterpriseItanium64 - Output - Boolean - True if operating system is Windows XP 64-bit Enterprise Itanium

bDatacenter64 - Output - Boolean - True if operating system is Windows XP 64-bit Datacenter

bEnterprise64 - Output - Boolean - True if operating system is Windows XP 64-bit Enterprise

bStandard64 - Output - Boolean - True if operating system is Windows XP 64-bit Standard

bComputeServer64 - Output - Boolean - True if operating system is Windows Compute Server 64-bit

bDatacenterServer64 - Output - Boolean - True if operating system is Windows 2000 Datacenter Server 64-bit

bEnterpriseServer64 - Output - Boolean - True if operating system is Windows

Enterprise Server 64-bit

  bWebBladeServer64 - Output - Boolean - True if operating system is Windows Blade Server 64-bit

  bStandardServer64 - Output - Boolean - True if operating system is Windows Standard Server 64-bit

```
================================================================================
Available in cFileDate (clsFileDate)
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  ' 01-Nov-2008 Reset order to match API SetFileTime()
  Public Enum enumDateProperties
      eCreateDate      ' 0
      eLastAccessed    ' 1
      eLastModified    ' 2
      eTwoDates        ' 3  Last Accessed, Last Modified (most common)
      eThreeDates      ' 4  Create Date, Last Accessed, Last Modified
  End Enum

' ************************************************************************
' ****                        Properties                              ****
' ************************************************************************
    StopProcessing - Input/Output - Boolean - True if user wants to stop processing

    SelectedDateField - Input only - Long Integer - Based on enumDateField values

    PathFileName - Input/Output - String - Full path and file name

    TimeStamp - Input/Output - Date - Date value of the time stamp (Created,
                Modified, last Accessed)

    LongDateFormat - Output only - String - Designates what format to use to
                display the date in long format in this users locale

    ShortDateFormat - Output only - String - Designates what format to use to
                display the date in short format in this users locale

    TimeFormat - Output only - String - Designates what format to use to
                display the time in this users locale

    DateSeparator - Output only - String - Designates which symbol to use to
                separate the short date display in this users locale

    TimeSeparator - Output only - String - Designates which symbol to use to
                separate the time display in this users locale

    CreateDate - Output only - Date - Folder or file creation date

    LastAccessed - Output only - Date - Folder or file last accessed date

    LastModified - Output only - Date - Folder or file last modified date

' ************************************************************************
' ****                        Methods                                 ****
' ************************************************************************
Format a date/time string based on the format desired by the user obtained from
the current system date/time settings.
Function SystemDateInfo(ByRef strDate As String, _
                        ByRef strTime As String, _
                Optional ByVal blnUseShortdate As Boolean = True) As Boolean

Format a date/time string based on the format desired by the user obtained from
the passed filename date/time settings.
Function FileDateInfo(ByRef strDate As String, _
                      ByRef strTime As String, _
```

```
            Optional ByVal blnUseShortdate As Boolean = True, _
            Optional ByVal lngDateField As enumDateProperties = eLastModified) As
Boolean
```

Change a folder or file date/time stamp based on date stored in property TimeStamp().

```
Function SetDateProperty() As Boolean
```

```
================================================================================
Available in cBigFiles (clsBigFiles)
================================================================================

' ************************************************************************
' ****                       Properties                               ****
' ************************************************************************
    StopProcessing - Input/Output - Boolean - True if user wants to stop processing


' ************************************************************************
' ****                       Methods                                  ****
' ************************************************************************
' Open a file to be used as input.  The file must already exist.
' If the file does not exist, an error will occur.
Function OpenReadOnly(ByVal strFileName As String, _
                      ByRef hFile As Long) As Boolean

' Open a file to update.  If the file exist, if will be opened.  If the
' file does not exist, it will be created.  Use carefully.  If you open
' an existing file and something goes wrong, the file may become a zero
' byte file.  There is no recovery of the data available. I use this to
' access a temporary work file only.
Function OpenReadWrite(ByVal strFileName As String, _
                       ByRef hFile As Long) As Boolean

' This routine is used to open a file as read only and calculate it's size.
Sub CalcFileSize(ByVal strFileName As String, _
                 ByRef curFilesize As Currency, _
         Optional ByRef strBitsInHex As String = "")

' This routine is used to read data from an opened file.
Function API_ReadFile(ByVal hFile As Long, _
                      ByVal curPosition As Currency, _
                      ByRef abytData() As Byte) As Boolean

' This routine is used to write data to the file.
Function API_WriteFile(ByVal hFile As Long, _
                       ByVal curPosition As Currency, _
                       ByRef abytData() As Byte) As Boolean

' Sets the pointer to the end of the file designating that we are now
' finished with this file.
Sub API_SetEndOfFile(ByVal hFile As Long, _
                     ByVal curPosition As Currency)

' Closes an open file.
Sub API_CloseFile(ByRef hFile As Long)

' Creates a file from 1 byte to greater than 2gb filled with null values.
' I have created files greater than 5gb without any problems.
Function CreateBigFile(ByVal strFileName As String, _
                       ByVal curFilesize As Currency) As Boolean

' Updates a file from 1 byte to greater than 2gb with null values.
Function LoadWithNullValues(ByVal hFile As Long, _
                            ByVal curFilesize As Currency) As Boolean
```

```
================================================================
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