CryptoAPI Hash Demo   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.
http://www.microsoft.com/downloads/details.aspx?FamilyId=7B9BA261-7A9C-43E7-9117-F673077FFB3C

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.

```
*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
*** 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:

The Cryptography API, or How to Keep a Secret
http://msdn.microsoft.com/en-us/library/ms867086.aspx

CryptoAPI Cryptographic Service Providers
http://msdn.microsoft.com/en-us/library/bb931357(VS.85).aspx

SHA-2 support on Windows XP
Paraphrasing: Regarding SHA-224 support, SHA-224 offers less security than SHA-256 but takes the same amount of resources. Also SHA-224 is not generally used by protocols and applications. The NSA's (National Security Agency) Suite B standards also do not include it. Microsoft has no plans to add it to future versions of their CSPs (Cryptographic Service Providers).
http://blogs.msdn.com/b/alejacma/archive/2009/01/23/sha-2-support-on-windows-xp.aspx

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 Feb-2011 (will supercede FIPS 180-3)
http://csrc.nist.gov/publications/drafts/fips180-4/Draft-FIPS180-4_Feb2011.pdf

WARNING:

MD4 Message-Digest Algorithm has 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/toolkit/secure_hashing.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

****

How to use:

For a simple example, execute the SHA_Demo application. The demo converts
the data to a byte array prior to passing it to the DLL to be hashed.

[STRING DATA]
Convert string data to byte array prior to passing to the HashString function.

```
Example:   abytData() = StrConv("abc", vbFromUnicode)


[FILE DATA]
Just the path and filename are passed in the byte array.  Convert the
path\filename data to byte array prior to passing to the HashFile function.
The HashFile routine will open and read the file into an internal byte array.

Example:
  abytData() = StrConv("C:\Files\Test Folder\Testfile.doc", vbFromUnicode)


Both will create a hashed output string based on file data input.

*****************************************************************************


=============================================================================
                              Overview
=============================================================================

' ***************************************************************************
' Enumerations
' ***************************************************************************
  Public Enum enumAPI_HashAlgorithms
      eAPI_MD2      ' 0
      eAPI_MD4      ' 1
      eAPI_MD5      ' 2
      eAPI_SHA1     ' 3
      eAPI_SHA256   ' 4
      eAPI_SHA384   ' 5
      eAPI_SHA512   ' 6
  End Enum

' ***************************************************************************
' ****                      Properties                                   ****
' ***************************************************************************
  Version - Output - String - Name of DLL and version information

  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)

  HashMethod - Input only - [OPTIONAL] Long integer (0-6) designating what
            hash algorithm to use. See enumHASH_ALGORITHM

  HashRounds - Input only - [OPTIONAL] Long integer (1-10) designating how
            many iterations of hashing the data are to be performed.
            Default = 1

  ReturnHexString - Input only - [OPTIONAL] Boolean
            True - Return hashed data as hex string.
            False - Return raw hashed data.

  AES_Ready - Output only - Boolean
            True - AES (Advanced Encryption Standard) is available (SHA2 family)
            False - Only MD2, MD4, MD5 and SHA-1 are available



' ***************************************************************************
' ****                      Methods                                      ****
' ***************************************************************************
' Creates a hash output string based on string data input.
Function HashString(ByRef abytInput() As Byte) As Variant

' Just the path and filename are passed in the byte array.
' The HashFile routine will open and read the file into
' another byte array prior to performing the hash.
Function HashFile(ByRef abytInput() As Byte) As Variant
```