kiCrypt.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 SP3 64-bit through Windows 10.
Windows XP 32-bit, 9x, 2000 and NT4 are no longer supported.

All algorithms, with the exception of Base64, 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:

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 MS Windows
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 does not include it. Microsoft
has no plans to add it to future versions of their Cryptographic
Service Providers (CSP).

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 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
http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA2_Additional.pdf

Aaron Gifford's additional test vectors
http://www.adg.us/computers/sha.html

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

****

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.

  Ex: 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.

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

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

Test data provided to test either hash or cipher:

    TestPhrase.txt ASCII text phrase   (Copy & paste phrase for string test)
    TestFile.txt   ASCII text file

  Binary test files:
    kB_32.dat   32,768 binary zeros
    OneMil_0.dat   One million binary zeros (FIPS 180-3)
    OneMil_a.dat   One million letter "a"   (FIPS 180-2)
    API32.txt   Text file over 1 MB

****

Note from Mark Hutchinson's presentation about Microsoft's VB random number
generator.   http://www.15seconds.com/issue/051110.htm

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 to re-initialize the Rnd function, 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

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/

****
** Enhanced ciphers
****

With all ciphers, except ArcFour, the data length will change. After encrytption, data sizes will not match original sizes. This is due to internal padding and the storing of information required to decrypt the data later.

****
** PASSWORDS
****

Currently there is a minimum and maximum length of the password the user may enter. This can be changed in the kiCrypt DLL basCommon.bas module. In the declarations section, locate these two constants and make the desired change. Be sure to recompile the DLL and the demo application.

```
PWD_LENGTH_MIN = 8
PWD_LENGTH_MAX = 50
```

If no hash algorithm is selected then the default hash will be SHA-256.

```
================================================================================
Available in Cipher (clsCipher)
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  Public Enum enumCIPHER_ALGORITHM
      CIPHER_ARCFOUR     ' 0
      CIPHER_BASE64      ' 1
      CIPHER_BLOWFISH    ' 2  Default
      CIPHER_GOST        ' 3
      CIPHER_RIJNDAEL    ' 4
      CIPHER_BLOWFISH    ' 5
      CIPHER_SKIPJACK    ' 6
      CIPHER_TWOFISH     ' 7
  End Enum


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

CipherMethod - Input - Long integer - (0-7) designating which cipher
          algorithm to use.  See enumCIPHER_ALGORITHM

Version - Output - String - DLL version information

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

Password - Input - String - Representing a user defined password.
      Syntax:  X.Password = "kenaso@tx.rr.com"

CreateNewFile - Input - Boolean - Used to desiginate if input file is to be
      overwritten after encryption/decryption.
      TRUE - Create new file to hold encrypted/decrypted data
      FALSE - Overwrite input file after encryption/decryption

KeyLength - Input - Long Integer

BlockSize - Input - Long Integer - Only used by Rijndael cipher

CipherRounds - Input - Long Integer - Number of times to perform
      encryption on some data

PrimaryKeyRounds - Input - Long Integer - Number of rounds to mix primary key
      array.  Only used by Blowfish, GOST. Serpent and Twofish ciphers.

HashMethod - Input - Long Integer - See enumHASH_ALGORITHM

CipherMethod - Input - Long Integer - See enumCIPHER_ALGORITHM

PasswordLength_Min - Output - Long Integer - Minimum password length

PasswordLength_Max - Output - Long Integer - Maximum password length


' ************************************************************************
' ****                        Methods                                 ****
' ************************************************************************
' Encrypt a string of data.
Function EncryptString(ByVal strData As String) As String
```

```
' Decrypt a string of data.
Function DecryptString(ByVal strData As String) As String


NOTE:  strSource = Path\Filename location
       strTarget  = Path\Filename location

' Encrypt a file.  If no target name is given then the process
' will use the source name.
Function EncryptFile(ByVal strSource As String, _
            Optional ByVal strTarget As String = "") As Boolean

' Decrypt a file.  If no target name is given then the process
' will use the source name.
Function DecryptFile(ByVal strSource As String, _
            Optional ByVal strTarget As String = "") As Boolean

' Convert a hex string, stored in a byte array, into a normal
' string of data, also stored in a byte array.
Sub HexToString(ByRef abytData() As Byte)

' Convert a normal string, stored in a byte array, into a hex
' string of data, also stored in a byte array.
Sub StringToHex(ByRef abytData() As Byte)
```

```
================================================================================
Located in Hash (clsHash.cls)
================================================================================

' ************************************************************************
' Enumerations
' ************************************************************************
  Public Enum enumHASH_ALGORITHM
      eHASH_MD2            ' 0
      eHASH_MD4            ' 1
      eHASH_MD5            ' 2
      eHASH_SHA1           ' 3
      eHASH_SHA256         ' 4  Default
      eHASH_SHA384         ' 5
      eHASH_SHA512         ' 6
      eHASH_TIGER128       ' 7
      eHASH_TIGER160       ' 8
      eHASH_TIGER192       ' 9
      eHASH_TIGER224       ' 10
      eHASH_TIGER256       ' 11
      eHASH_TIGER384       ' 12
      eHASH_TIGER512       ' 13
      eHASH_WHIRLPOOL224   ' 14
      eHASH_WHIRLPOOL256   ' 15
      eHASH_WHIRLPOOL384   ' 16
      eHASH_WHIRLPOOL512   ' 17
  End Enum

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

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

HashMethod - Input only - [OPTIONAL] Long integer (0-17) designating what
     hash algorithim to use. See enumHASH_ALGORITHM

HashRounds - Input only - [OPTIONAL] Long integer (1-10) designating number
     of hash iterations to use.

ReturnLowercase - Input only - [OPTIONAL] Boolean - Return hashed data in
     upper or lower case format.  Default = TRUE

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

' Creates a hash output string based on file data input.
' Input is the Path/File location only in byte array format.
Function HashFile(ByRef abytInput() As Byte) As Variant
```

```
================================================================================
Available in CRC32 (clsCRC32.cls)
================================================================================

' ************************************************************************
' ****                        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)


' ************************************************************************
' ****                        Methods                                   ****
' ************************************************************************
' Returns a numeric value representing a string input.
Function CRC32_String(ByVal strData As String) As String

' Returns a numeric value representing a Path/File location only.
Function CRC32_File(ByVal strSource As String) As String
```

```
================================================================================
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 type structures,
' numbers with decimal values, etc.
Sub SwapData(ByRef vntData1 As Variant, _
             ByRef vntData2 As Variant)
```

```
' Swap string data with each other.  Uses API CompyMemory() function
' to exchange address pointers.  Very fast.
Sub SwapStrings(ByRef strData1 As String, _
                ByRef strData2 As String)

' 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 SwapBytes(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

' Capture millisecond count to be used in calculating a seed value for
' Visual BASIC Random Number Generator.  Manipulate captured data and
' return a long integer.
Function GetTmpSeedValue() As Long

' This routine will capture current number of milliseconds.
Function CurrentMilliseconds() As Currency

' Calculates and formats elapsed time.
' Ex:  12:34:56.7890   High resolution performance timer available
'      12:34:56.789    No high resolution performance timer available
Function ElapsedTime(ByVal curStart As Currency, _
                     ByVal curFinish As Currency) As String

' Encrypt data string with the CryptoAPI for the purpose of creating
' random data only.
Function RC4_Encrypt(ByRef abytData() As Byte, _
            Optional ByVal strData As String = vbNullString, _
            Optional ByVal blnUniquePwd As Boolean = True) As Boolean
```

License   Kenneth Ives kenaso@tx.rr.com
                                 Copyright © 2004-2016
                                 All rights reserved

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.