kiHash DLL 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.
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.

*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
*** 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
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

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

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

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 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

**********

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 Hash modules
*******************************

Convert string data to byte array prior to passing to the HashString function.

Ex: abytData() = StrConv("abc", vbFromUnicode) ' string to byte array


Convert the path\filename data to byte array prior to passing to the Hashfile
function.

Ex: abytData() = StrConv("C:\Program Files\Test folder\Testfile.doc", vbFromUnicode)


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

Module:        clsMD4

Description:   This software is provided ``AS IS'' and any express or
               implied warranties, including, but not limited to, the
               implied warranties of merchantability and fitness for a
               particular purpose, are disclaimed.  In no event shall the
               authors be liable for any direct, indirect, incidental,
               special, exemplary, or consequential damages (including, but
               not limited to, procurement of substitute goods or services;
               loss of use, data, or profits; or business interruption)
               however caused and on any theory of liability, whether in
               contract, strict liability, or tort (including negligence
               or otherwise) arising in any way out of the use of this
               software, even if advised of the possibility of such damage.

               This is a class which encapsulates a set of MD5 Message
               Digest functions.  MD5 algorithm produces a 128 bit digital
               fingerprint (signature) from an dataset of arbitrary length.
               For details see RFC 1321 (summarized below).  This
               implementation is derived from the RSA Data Security, Inc.
               MD5 Message-Digest algorithm reference implementation
               (originally written in C).

NOTES:
     Network Working Group                                    R. Rivest
     Request for Comments: 1320        MIT Laboratory for Computer Science
     Obsoletes: RFC 1186                        and RSA Data Security, Inc.
                                                             April 1992


                          The MD4 Message-Digest Algorithm

     Summary

        This document describes the MD4 message-digest algorithm [1]. The
        algorithm takes as input a message of arbitrary length and produces
        as output a 128-bit "fingerprint" or "message digest" of the input.
```

It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD4 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD4 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD4 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly.

RFC Author:
Ronald L.Rivest
Massachusetts Institute of Technology
Laboratory for Computer Science
NE43 -324545    Technology Square
Cambridge, MA  02139-1986

**********

Module:  clsMD5


This is a class which encapsulates a set of MD5 Message Digest functions.  MD5 algorithm produces a 128 bit digital fingerprint (signature) from an dataset of arbitrary length. For details see RFC 1321 (summarized below).  This implementation is derived from the RSA Data Security, Inc. MD5 Message-Digest algorithm reference implementation (originally written in C).

NOTES:

Network Working Group  R. Rivest
Request for Comments: 1321  MIT Laboratory for Computer Science
and RSA Data Security, Inc.
April 1992

The MD5 Message-Digest Algorithm

Summary

This document describes the MD5 message-digest algorithm. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any

message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly.

The MD5 algorithm is an extension of the MD4 message-digest algorithm 1,2]. MD5 is slightly slower than MD4, but is more "conservative" in design. MD5 was designed because it was felt that MD4 was perhaps being adopted for use more quickly than justified by the existing critical review; because MD4 was designed to be exceptionally fast, it is "at the edge" in terms of risking successful cryptanalytic attack. MD5 backs off a bit, giving up a little in speed for a much greater likelihood of ultimate security. It incorporates some suggestions made by various reviewers, and contains additional optimizations. The MD5 algorithm is being placed in the public domain for review and possible adoption as a standard.

RFC Author:
Ronald L.Rivest
Massachusetts Institute of Technology
Laboratory for Computer Science
NE43 -324545    Technology Square
Cambridge, MA  02139-1986

RSA Copyright Notice



**********
Project:  RipeMD128 and RipeMD160
Module:  clsRipeMD128_256.cls and clsRipeMD160_320.cls

Description:  RipeMD is an iterative hash function that operates on 32-bit words.  The round function takes as input a 10-word chaining variable and a 16-word message block and maps this to a new chaining variable. All operations are defined on 32-bit words. Padding is identical to that of MD4.

RIPEMD-256 and RIPEMD-320 are optional extensions of,

respectively, RIPEMD-128 and RIPEMD-160, and are intended for applications of hash functions that require a longer hash result without needing a larger security level. If you require a stronger hash, I recommend you use a member of the SHA2 family of algorithms.

**********

Copyright Information



**********

Project: Secure Hash Algorithm
SHA-1 SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256

Description: The Secure Hash Algorithm (SHA) is required for use with the Digital Signature Algorithm (DSA) as specified in the Digital Signature Standard (DSS) and whenever a secure hash algorithm is required for federal applications. For a message of length < 2^64 bits, this algorithm produces a condensed representation of the message called a message digest. The message digest is used during generation of a signature for the message. This also used to compute a message digest for the received version of the message during the process of verifying the signature. Any change to the message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify.

These algorithms have been tested to be accurate in accordance

with FIPS 180-2, FIPS 180-3, FIPS 180-4 publications. Also, test vectors by Aaron Gifford at:
http://www.adg.us/computers/sha.html

According to FIPS-180-2 there are only two differences between SHA-224 and SHA-256.

1. The initalizing values are different
2. Just the left most 224 bits (28 bytes) are saved

According to FIPS-180-2 there are only two differences between SHA-384 and SHA-512.

1. The initalizing values are different
2. SHA-384 only uses the first six elements for the output. SHA-512 uses all eight elements for the output.

IMPORTANT NOTE
I have kept SHA-1 in this DLL because I have other uses for a fast hash that will not compromise security. (i.e., Seeding or enhancing random number generation)

**********

Module: Tiger3 Hash Algorithm

This module is my experimental version for Tiger-128 thru Tiger-512 bit output. I call it Tiger3. I have not had any problems with it thus far. If you should encounter any problems, please email me at:

Kenneth Ives  kenaso@tx.rr.com

DO NOT CONTACT  Ross Anderson, http://www.cl.cam.ac.uk/users/rja14/
Eli Biham, http://www.cs.technion.ac.il/~biham

because they did not write this module nor are they responsible in any manner as to its content.

Description: Tiger3 is a fast New cHash function, designed to be very fast on modern computers, and in particular on the state-of-the-art 64-bit computers (like DEC-Alpha), while it is still not slower than other suggested hash functions on 32-bit machines.

Tiger hash has no usage restrictions nor patents. It can be used freely, with the reference implementation, with other implementations or with a modification to the reference implementation (as long as it still implements Tiger2). We only ask you to let us know about your implementation and to cite the origin of Tiger and of the reference implementation.
http://www.cs.technion.ac.il/~biham/Reports/Tiger/

Special Note: I decided to make changes to initial work data arrays [malngHash()]. I believe that this is a cleaner and more secure method as to the calculation of the Tiger hashes. See Initialize() routine for more details.

Reference: A Fast New cHash function
http://www.cs.technion.ac.il/~biham/Reports/Tiger/

Original authors of the Tiger hash:
Ross Anderson, http://www.cl.cam.ac.uk/users/rja14/
Eli Biham, http://www.cs.technion.ac.il/~biham

Found 32-bit C source code at:
http://www.cs.technion.ac.il/~biham/Reports/Tiger/tiger-src32.zip

**********

Module: Whirlpool Hash

Whirlpool versions for -224, -256, -384 bit output are my experiment. Any problems with these outputs, please email me at:

Kenneth Ives kenaso@tx.rr.com

DO NOT CONTACT Vincent Rijmen
Paulo S. L. M. Barreto

because they did not write this module nor are they responsible in any manner as to its content.

Description: The WHIRLPOOL Hashing Function
Designed by Vincent Rijmen and Paulo S. L. M. Barreto

Whirlpool is a cryptographic hash function adopted by the International Standards Organization (ISO) and International Electrotechnical Commission (IEC) as part of the joint ISO/IEC 10118-3 international standard. It takes an arbritrary block of data and returns a 512 bit digest that can be used as a digital fingerprint for message authentication. Compliance with the standard may be verified at:
http://hash.online-convert.com/whirlpool-generator

Whirlpool is a hash designed after the Square block cipher. Whirlpool is a Miyaguchi-Preneel construction based on a substantially modified Advanced Encryption Standard (AES). It takes a message of any length less than 2256 bits and returns a 512-bit message digest.

The authors have declared that "WHIRLPOOL is not (and will never be) patented and may be used free of charge for any purpose. The reference implementations are in the public domain."

The algorithm is named after the Whirlpool Galaxy in Canes Venatici.

Using the reference C implementation on a 1 GHz Pentium III platform, we observe that Whirlpool operates at about 73 cycles per hashed byte.

The compression function runs at about 56 cycles per hashed byte. Many factors explain the observed performance. First, a 32-bit processor was used to test a native 64-bit implementation; better results are expected by merely running the speed measurement on an Alpha or Itanium processor. Second, it seems that the pipe parallelism capabilities of the Pentium were not fully used; this may reflect a non-optimising implementation of 64-bit arithmetic support by the C compiler, and might be overcome by an assembler implementation. Third, the tables employed in the reference implementation are quite large, and the built-in processor cache might not be enough to hold

them, the data being hashed, and the hashing code at once, thus degrading processing speed.

Whirlpool is much more scalable than most modern hashing functions. Even though is not specifically oriented toward any platform, it is rather efficient on many of them, its structure favouring extensively parallel execution of the component mappings. At the same time, it does not require excessive storage space (either for code or for tables), and can therefore be efficiently implemented in quite constrained environments like smart cards, although it can benefit from larger cache memory available on modern processors to achieve higher performance. It does not use expensive or unusual instructions that must be built in the processor. The mathematical simplicity of the primitive resulting from the design strategy tends to make analysis easier. And finally, it has a very long hash length; this not only provides increased protection against birthday attacks, but also offers a larger internal state for entropy containment, as is needed for certain classes of pseudo-random number generators.

Reference:  The WHIRLPOOL Hash Function (Home page)
http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html

Whirlpool (cryptography)
http://en.wikipedia.org/wiki/Whirlpool_(cryptography)

Whirlpool Hashing Function in Visual Basic 6.0
Author:  Korejwa2   12/31/2010

http://www.Planet-Source-Code.com/vb/scripts/ShowCode.asp?txtCodeId=73638&lngWId=1

**********

==========

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