MarshallSoft AES

(Advanced Encryption Standard)

Library for Visual Basic

Programmer's Manual

(AES4VB)

Version 6.0

February 23, 2022

This software is provided as-is.
There are no warranties, expressed or implied.

MarshallSoft Computing, Inc.
Post Office Box 4543
Huntsville AL 35815

Email: info@marshallsoft.com
Web: www.marshallsoft.com

TABLE OF CONTENTS

1 Introduction

The **MarshallSoft Advanced Encryption Standard Library for Visual Basic (AES4VB)** is a toolkit that allows software developers to easily implement strong encryption and decryption into a Windows Visual Basic Visual Basic or Visual Basic .NET application.

The **MarshallSoft Advanced Encryption Standard Library (MarshallSoft AES)** is a component (DLL) library of functions used to perform encryption and decryption using the 256-bit "Advanced Encryption Standard" (AES) as specified by the U.S. National Institute of Standards and Technology (NIST). See http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

AES is considered "strong encryption" and replaces the previous U.S. encryption standard "Data Encryption Standard" (DES). AES is commonly used by many financial entities such as banks to protect their customer's sensitive information.

Our implementation of the Advanced Encryption Standard (AES) has been verified by running the "Advanced Encryption Standard Algorithm Validation Suite" (AESAVS), which can be found at http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.

This **MarshallSoft Advanced Encryption Standard Programmers Manual for Visual Basic** provides information need to compile and run programs in a Visual Basic programming environment.

The **MarshallSoft Advanced Encryption Standard DLL's** will work under all versions of Windows (2003-2012/XP/Vista /Windows7/Windows 8). Both Win32 and Win64 DLL's are included.

AES4VB includes several Visual Basic example programs, including Visual Basic and Visual Studio, which demonstrate AES encryption and decryption. Borland C++ Builder (BCB) examples are also provided.

The **MarshallSoft Advanced Encryption Standard Library for Visual Basic** component library supports and has been tested with Microsoft Visual Basic (VB 4.0 – VB 6.0), Microsoft Visual Studio .NET Framework and Microsoft Visual Studio through Visual Studio 2022. **AES4VB** can also be used with any VBA (Visual Basic for Applications) language such as Excel, Access, MS Office, etc. as well as with PowerBuilder.

The **MarshallSoft AES** DLLs (AES32.DLL and AES64.DLL) can also be used from any language (C/C++, .NET, C#, Borland/Embarcadero Delphi, Visual FoxPro, COBOL, Xbase++, Visual dBase, Microsoft Office, etc.) capable of calling the Windows API.

For the latest version of the **MarshallSoft AES** software, see www.marshallsoft.com/aes.htm.

Legalities

It is illegal to possess strong encryption software in some countries in the world. Do not download or use this software if it is illegal to do so in your country.

In addition, this software cannot be sold to countries on the U.S. Embargo List. See http://www.pmddtc.state.gov/embargoed_countries/index.html

1.1 Features

Some of the many features of the **Advanced Encryption Library (AES)** component library are as follows:

- Works with 32-bit and 64-bit Windows.
- Implements the 256-bit "**Advanced Encryption Standard**" (Rijndael)
- Implements Diffie-Hellman key exchange.
- Supports **ECB** (Electronic Cookbook) mode.
- Supports **CBC** (Cipher Block Chaining) mode.
- Supports SHA-256 cryptographic hash algorithm.
- **Free** technical support and updates for one year.
- License covers all programming languages.
- Royalty free distribution with a compiled application.
- Evaluation versions are fully functional. (30 day trial). No unlock code is required.
- Can be used from GUI mode or console mode programs.
- Is fully thread safe.
- Supports 32-bit and 64-bit Windows through Windows 11.
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Both Win32 and Win64 DLLs are included.
- Is native Windows code but can also be called from managed code.
- Will run on machines with or without .NET installed.
- Works with all 32-bit versions of Microsoft Visual Basic.
- Works with all 32-bit and 64-bit versions of Microsoft Visual Studio.
- Works with VBA (Visual Basic for Applications) such as Excel or Access as well as Power Builder.
- Does **not** depend on support libraries. Makes calls to core Windows API functions only.
- Can be used with any program (in any language) capable of calling Windows API functions such as C/C++, C#, Visual FoxPro, Delphi, Xbase++, dBASE or COBOL.
- Can be purchased with (or without) source code.
- Updates are free for one year (source code updates are separate).
- Unlimited one-year email and phone tech support.
- Documentation online as well as in printable format.

A selection of Visual Basic example programs with full source code is included. Refer to Section 4 for more details on each of the example programs.

```
1. aesver     Displays AES4VB version
2. TestAES    Performs AES encryption / decryption tests
3. Crypto     Encrypts and/or decrypts a file
4. Password   Manages passwords kept encrypted on disk.
5. Validate   Validates AES against on-line encryption.
6. TestDH     Tests Diffie-Hellman key exchange.
```

1.2 Documentation Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (AES_4VB) in the set.

- AES4VB Programmer's Manual (AES_4VB.PDF)
- AES User's Manual (AES_USR.PDF)
- AES Reference Manual (AES_REF.PDF)

The AES_4VB Programmer's Manual (AES_4VB.PDF) is the language specific (Visual Basic) manual. All Visual Basic programming issues such as compiling, compilers and example programs are discussed in this manual.

The AES User's Manual (AES_USR.PDF) discusses SMTP and POP3/IMAP email processing as well as language independent programming issues such as application notes. Purchasing and licensing information is also provided.

The AES Reference Manual (AES_REF.PDF) contains details on each individual AES function and provides a list of AES error codes.

The online documentation can be accessed on the **MarshallSoft Advanced Encryption Library (AES) for Visual Basic** product page at:

http://www.marshallsoft.com/aes4vb.htm

1.3 Example Program

The following example demonstrates the use of some of the **MarshallSoft Advanced Encryption Library (AES) for Visual Basic** component library functions:

```
Private Function EncryptFile(ByVal KeyBuffer As String,
                             ByVal InFile As String,
                             ByVal OutFile As String) As Integer
Dim Code As Integer
Dim Control As String
Dim Vector As String
' vector not used in ECB mode
Vector = Chr(0)
Control = "*"
' initialize for encryption in ECB mode
Code = aesInitAES(KeyBuffer, Vector, AES_ECB_MODE, AES_ENCRYPT, Control)
If Code < 0 Then
  EncryptFile = Code
  Exit Function
End If
' encrypt the file (InFile -> OutFile)
Code = aesEncryptFile(Control, InFile, OutFile)
EncryptFile = Code
End Function
```

1.4 Installation

(1) Before installation of AES4VB, your Visual Basic compiler (any version) should already be installed on your system and tested.

(2) Unzip AES4VB60.ZIP (evaluation version) or AESxxxxx.ZIP (registered version where xxxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program SETUP.EXE that will install all AES4VB files, including copying AES32.DLL and AES64.DLL to the Windows directory.

VB 4.0, VB 5.0, and VB 6.0 project filenames end with the extension ".vbp", and VB.NET/Visual Studio .NET project filenames end with "vbproj". For example,

```
AESVER.VBP    --- Project file for 32-bit Visual Basic (VB_4.0, 5.0, 6.0)
AESVER.VBPROJ --- Project file for VB.NET / Visual Studio
```

Note that no DLL registration is required.

1.5 Uninstalling

Uninstalling AES4VB is very easy. First, run UINSTALL.BAT, which will delete AES32.DLL and AES64.DLL from the Windows directory, typically C:\WINDOWS. Second, delete the AES project directory created when installing AES4VB.

1.6 Pricing

A developer license for the **MarshallSoft Advanced Encryption Standard Library** can be purchased for $119 (or $199 with source code to the library DLL). Purchasing details can be found in Section _1.4, "How to Purchase", of the AES User's Manual (AES_USR).

Also see INVOICE.TXT or http://www.marshallsoft.com/order.htm

Registration includes one year of free updates and technical support. Registered DLLs never expire.

1.7 Updates

When a developer license is purchased for AES, the developer will be sent a set of registered DLLs plus a license file (AESxxxxx.LIC). The license file can be used to update the registered DLL for a period of one year from purchase. Updates can be downloaded from

http://www.marshallsoft.com/update.htm

After one year, the developer license must be updated to be able to download updates and receive technical support. The license can be updated for:

- $33 if the update is ordered within one year of the original purchase (or previous update).
- $55 if the update is ordered between one and three years of the original purchase (or previous update).
- $77 if the update is ordered after three years of the original purchase (or previous update). .

The update price includes technical support for an additional year. Note that the registered DLLs, (AES32. DLL and AES64.DLL) never expire. If source code was previously purchased, updates to the source code can be purchased for $40 along with the license update.

2 Library Overview

The **Advanced Encryption Library (AES)** component library has been tested on multiple computers running Windows 2003-2012/XP/Vista/Windows 7/Windows 8 (Win32/Win64).

The AES4VB library has been tested with several Visual Basic compilers, from VB 4.0 through VB 6.0, Microsoft Visual Basic .NET (**VB.Net**.) and Microsoft Visual Studio (2005- 2015). AES can also be used with any VBA language such as Excel, Access, MS Office as well as PowerBuilder.

The SETUP installation program will copy the DLL's to the Windows directory. Refer to Section 1.4 "Installation".

After SETUP is run, the AES4VB files are copied to the directory specified (default \AES4VB). Three sub-directories are created, as follows:

```
DOCS – All documentation files
APPS – All example code
DLLS – All DLL's
```

2.1 Dynamic Link Libraries

The **Advanced Encryption Library (AES)** component library SDK uses a Win32 [AES32.DLL] and Win64 [AES64.DLL] dynamic link library (DLL). A DLL is characterized by the fact that it need not be loaded until required by an application program and that only one copy of the DLL is necessary regardless of the number of application programs that use it. Contrast this to the traditional static library that is bound to each and every application that uses it at link time.

An important advantage that DLL's have over other "popular" library formats such as VBX or OCX is that DLL's are callable by all Windows applications. Since DLL's are the building blocks of the Windows Operating System, they will not be replaced by a "newer technology".

The following files can be found in the DLL sub-directory when SETUP is run:

```
aes32.dll   - Win32 version of AES
aes64.dll   - Win64 version of AES
```

2.2 Keycode

AES32.DLL and AES64.DLL are encoded with a keycode. The keycode is a 9 or 10 digit decimal number (unless it is 0), and will be found in the file KEYCODE.BAS (and KEYCODE.VB). The keycode for the evaluation version is 0. The developer will receive a new keycode and AES32.DLL after registering. The KEYCODE is passed to **aesAttach**.

If you get an error message (value -74) when calling **aesAttach**, it means that the keycode in your application does not match the keycode in the DLL. After registering, it is best to remove the evaluation version of the AES32.DLL and AES64.DLL from the Windows search path or delete them.

2.3 Win32 / Win64 STDCALL and DECLSPEC

Our libraries are written in ANSI C and compiled using the _stdcall and _declspec keywords. This means that AES4VB uses the same calling conventions and file naming conventions as the Windows API

The AES library functions may be called by any Windows application program capable of calling the Windows API provided the proper declaration file is used.

2.4 Using Threads

AES4VB AES32.DLL and AES64.DLL) is thread safe and can be used from any Windows application capable of using threads.

2.5 Dynamic Strings

2.5.1 Passing VB Strings to DLL Functions

When passing a string to a DLL function, the DLL looks for a null character to terminate the string. This null character CHR(0) is normally present in Visual Basic strings, but can be lost if the string is modified by Visual Basic at runtime. This problem can be overcome by appending CHR(0) to the end of strings passed to **AES** functions.

2.5.2 Passing VB String Buffers to DLL Functions.

The Visual Basic language uses a technique known as "garbage collection" to manage string space at runtime, and may be called internally at any time by the Visual Basic runtime, asynchronous to what you may be doing in your code.

When passing a string buffer to a DLL function into which text will be copied, it is strongly recommended that the local string be allocated immediately before use. For example, a string buffer is passed to the dllGetMessage function, which copies a text message into it. Note that `SPACE(80)` is called immediately before dllGetMessage.

```
Dim Code As Integer
Dim Buffer As String * 80
' allocate buffer just before call to dllGetMessage
Buffer = SPACE(80)
' copy message into 'Buffer'
Code = dllGetMessage(Buffer, 80)
' message text is now in 'Buffer'
```

This technique is not necessary for passing a string to a DLL function, only when passing a buffer to a DLL into which data is to be placed by the DLL function.

2.6 Visual Studio (VB.NET)

There are a few differences between VB 4/5/6 and Visual Studio (VB.NET) that affect writing programs that use the **AES** library.

1. Variables that are declared “As Long” in VB 4/5/6 are declared “As Integer” in Visual Studio (VB.NET).

2. Fixed length strings are not supported in Visual Studio (VB.NET). When calling any **AES** function that can return a string, memory for the string variable must be allocated first. For example:

```
Buffer = Space(128)
Length = aesErrorText(0, ErrCode, Buffer, 128)
```

3. Some Visual Basic functions must be fully qualified. For example, instead of LEFT, use `Microsoft.VisualBasic.Left`

4. The module AES32.VB (not AES32.BAS) must be included in all VB.NET programs. For Win64 programs, include AES64.VB instead.

2.7 Visual Basic for Applications (VBA)

The **Advanced Encryption Library (AES)** component library can be used with Microsoft VBA applications such as EXCEL, ACCESS, and Microsoft Office.

Start EXCEL (or other 32-bit Office VBA program such as WORD or ACCESS), then enter design mode. Enable the "Controls Toolbox": choose "Tools" on the menu bar, then "Customize", and then check "Control Toolbox". From the control toolbox, choose and position a "Command Button". This will create code that looks like

```
Private Sub CommandButton1_Click()

End Sub
```

Replace the generated code with vbaTestAES.bas. The easiest way to do this is to paste from the clipboard.

Exit design mode and then press the command button to start this example program.

2.8 Power Builder

The **MarshallSoft Advanced Encryption Library (AES)** can also be used with Power Builder applications. Refer to PBUILDER.TXT in the \APPS subdirectory for more information.

`AES32.PBI` : Power Builder declaration file.

2.9 Adding AES4VB to a Project

Copy AES32.BAS (if running VB_4.0 /5/6), AES32.VB (if running Visual Studio Win32), or AES64.VB (if running Visual Studio Win64) to the same directory (folder) as the application program to which you want to add **AES** code. You will find these files in the \APPS sub-directory (folder) created when you ran SETUP

After the project settings are modified (see Sections 2.10.1 and 2.10.2 below), the first **AES** function that must be called is **aesAttach**. Often, this is best done when the Visual Basic form is first loaded. For example,

```
Private Sub Form_Load()
Dim Code As Integer
Code =aesAttach(AES_KEY_CODE, 0)
If Code < 0 Then
  MsgBox "ERROR: Cannot attach. Check AES_KEY_CODE."
  End
End If
End Sub
```

2.9.1 Adding AES4VB to a VB 4.0, 5.0, or 6.0 Project

Open the existing project with "File", "Open Project". Then choose "Insert", "Module", then add AES32.BAS and KEYCODE.BAS to your project. If prompted to add "DAO 2.50 Object Library", choose "no".

AES functions can now be called from your Visual Basic program.

2.9.2 Adding AES4VB to your VB.NET Project

Open the existing project with "File", "Open Project". Then choose "Project", "Add Module", then add AES32.VB and KEYCODE.VB to your project.

AES functions can now be called from your VB.NET program.

2.9.3 Adding AES4VB to your VS 2005, VS 2008, VS 2010, VS2012- 2022 Project

Open the existing project with "File", "Open Project". Then choose "Project", "Add Module", then add modules AES32.VB (or AES64.VB) and KEYCODE.VB to your project. If empty modules are created, replace the contents of these modules with the contents of the modules of the same name provided by us.

AES functions can now be called from your Visual Studio program.

2.10 Error Display

The error message text associated with **AES** error codes can be displayed by calling **aesErrorText**. Each sample program contains examples of error processing. Also refer to the file **aesErrors.txt** for a list of all AES error codes.

3 Compiler Issues

The **MarshallSoft Advanced Encryption Library (AES) for Visual Basic** component library supports and has been tested with all versions of Microsoft Visual Basic including:

- Visual Basic 4, 5, 6
- Visual Basic .NET (VB 7)
- Visual Basic .NET 2003 (VB 7.1)
- Visual Studio 2005 (VB 8.0)
  (through)
- Visual Studio 2022 (VB 17)

AES can also be used with any VBA language such as Excel, Access, MS Office as well as PowerBuilder.

3.1 Visual Basic Makefiles

The first Visual Basic for Windows (VB Version 3.0) uses a text file known as a "Visual Basic makefile" `(.MAK)` to list all the file components necessary to compile a program. Beginning with Visual Basic Version 4.0, the "Visual Basic Project file" `(.VBP)` was added. Both formats are "project files".

Beginning with Visual Studio 2003 (VB.Net), Visual Basic project files use extension ".vbproj".

3.2 Compiling Programs

Project files are provided for all examples. Project files for VB 4/5/6 end with the extension ".vbp" and end with ".vbproj" for Visual Studio (and VB.NET).

The example programs can be compiled from the Visual Basic development environment using the provided Visual Basic project files. Choose "File", then "Open Project" from the main VB menu.

After opening a project, VB v5.0 (and VB v6.0) users can save the project files in the VB v5.0 (or VB v6.0) format. When saving the example programs in VB v5.0 or VB v6.0 format, answer "no" if asked to add the "Microsoft DAO v2.5 library".

Compile and run AESVER as the first example to check your installation.

3.3 Explicitly Loading AES32.DLL / AES64.DLL

When an application program runs that makes calls to AES32.DLL (or AES64.DLL), the Windows operating system will locate AES32.DLL (or AES64.DLL), by searching the directories as specified by the Windows search path. If the AES32.DLL (or AES64.DLL), is placed in the \WINDOWS directory or \WINNT for Windows NT/2000, it will always be found by Windows.

AES32.DLL (or AES64.DLL), can be loaded from an explicit location by replacing "AES32.DLL" in AES32.BAS or AES32.VB by the full path. For example, to load AES32.DLL from C:\AES4VB\DLLS, the first entry in would be:

```
Declare Function aesAttach Lib "C:\AES4VB\DLLS\AES32.DLL" (ByVal KeyCode As
Integer, ByVal Flags As Integer) As Integer
```

The above also applies to AES64.DLL as well as AES32.DLL

4 Example Programs

Multiple Visual Basic example programs are included in **MarshallSoft AES Library for Visual Basic** (**AES4VB**). Examples for both Visual Basic 4/5/6 and Visual Studio (and VB.NET) are included.

Each example program comes with a VB project file.

- VB 4/5/6 project files end with ".VBP "
- Visual Studio / VB.Net project files end with ".VBPROJ".

Before writing your own programs, compile and run several of the example programs.

4.1 aesver

The AESVER example program displays the **AES** library version number and registration string. . Its purpose is to display the **AES** version, build, and registration string as well as to verify that AES32.DLL or AES64.DLL is being found and loaded by Windows.

```
Open project aesVer.vbp              -- VB 4, 5, 6
Open project aesVer.vbproj           -- Visual Studio (VB.NET)
Open project aesVer(VSxxxx).vbproj   -- Visual Studio xxx

     where xxxx = 2008,2010,2012,2013,2015,2017,1019,2022
```

4.2 TestAES

TestAES demonstrates how to encrypt and decrypt messages.

```
Open project TestAES.vbp             -- VB 4, 5, 6
Open project TestAES.vbproj          -- Visual Studio (VB.NET)
Open project TestAES(VSxxxx).vbproj  -- Visual Studio xxxx

     where xxxx = 2010,2012,2013,2015,2017,1019,2022
```

4.3 Crypto

Crypto demonstrates how to encrypt a file and to decrypt a (previously encrypted) file.

```
Open project Crypto.vbp              -- VB 4, 5, 6
Open project Crypto.vbproj           -- Visual Studio (VB.NET)
Open project Crypto(VSxxxx).vbproj   -- Visual Studio xxxx

     where xxxx = 2010,2012,2013,2015,2017,1019,2022
```

4.4 Password

Password manages a set of 5 passwords which are always kept encrypted on disk. A master password is used to access the set of 5 passwords.

```
Open project Password.vbp              -- VB 4, 5, 6
Open project Password.vbproj           -- Visual Studio (VB.NET)
Open project Password(VSxxxx).vbproj   -- Visual Studio xxxx

     where xxxx =2010,2012,2013,2015,2017,1019,2022
```

4.5 HashDigest

HashDigest computes the hash digest of a "salted" password.

```
Open project HashDigest.vbp              -- VB 4, 5, 6
Open project HashDigest.vbproj           -- Visual Studio (VB.NET)
Open project HashDigest(VSxxxx).vbproj   -- Visual Studio xxxx

     where xxxx = 2010,2012,2013,2015,2017,1019,2022
```

4.6 Validate

Validate tests against on-line encryption.

```
Open project Validate.vbp              -- VB 4, 5, 6
Open project Validate.vbproj           -- Visual Studio (VB.NET)
Open project Validate(VSxxxx).vbproj   -- Visual Studio xxxx

     where xxxx = 2010,2012,2013,2015,2017,1019,2022
```

4.7 TestDH

TestDH performs Diffie-Hellman key exchange test.

```
Open project TestDH.vbp              -- VB 4, 5, 6
Open project TestDH.vbproj           -- Visual Studio (VB.NET)
Open project TestDH(VSxxxx).vbproj   -- Visual Studio xxxx

     where xxxx = 2010,2012,2013,2015,2017,1019,2022
```

5 Revision History

The Advanced Encryption Library (AES) DLLs (AES32.DLL and AES64.DLL) are written in ANSI C. All language versions of AES (Visual Basic, Delphi, Visual Basic, PowerBASIC, FoxPro, Delphi, Xbase++, COBOL, and FORTRAN) use the same identical DLLs.

Version 1.0: March 30, 2013.

- Initial release.

Version 2.0: June 12, 2014

- Added aesEncryptWrite() function that encrypts data & writes to a file.
- Added aesReadDecrypt() function that reads an encrypted file & decrypts.
- Added aesSha256() function that computes the SHA-256 data hash.
- Added AES_SHA256_METHOD key generation method to aesMakeUserKey().
- Added Visual Studio 2012 and 2013 project files..
- Added PASSWORD example program.

Version 3.0: May 13, 2015

- Replaced function aesSha25() with aesSha256Data() and aesSha256File().
- Added PKCS7 padding option to aesPadBuffer().
- Added AES_PKCS7_MASK to “Flags” argument in aesAttach() to set file padding to PKCS7.

Version 4.0: November 21, 2016

- Added aesDecryptBuffer() that decrypts buffer of (encrypted) bytes.
- Added aesEncryptBuffer() that encrypts buffer of bytes.
- Added aesRemovePad() that removes PKCS7 padding.
- Added aesSaltPass() that generates ("salts") additional password characters.
- Added example program HashDigest that computes SHA 256 hash digest.

Version 4.1: June 27, 2017

- Fixed problem in aesMakeUserKey() using AES_SHA256_METHOD.
- Added AES_MIXED_METHOD method to aesMakeUserKey().
- Added aesSetInteger() and AES_SET_SEED that seeds the random number generator.
- Added aesShredFile() that shreds (overwrites with zeros then deletes) a file.

Version 4.2: July 6, 2018

- Added cryptograhically secure pseudo-random number generator aesSecureRandom().
- Added AES_GET_SECURE_SIZE to aesGetInteger().

Version 5.0: July 14, 2020

- Fixed problem in which AES_SET_SEED did not always reset the seed.
- Replaced deprecated function strncpy().
- Fixed internal problem with long (over 42 characters) pass phrases.
- Added function aesEncodeBase64 that Base64 encodes a data buffer.
- Added function aesDecodeBase64 that decodes a Base64 encoded data buffer.

Version 6.0: February 23, 2022

- Fixed problem handling large files in aesEncryptWrite() & aesReadDecrypt()
- Added Diffie-Hellman key exchange function aesMakeKeyPair()
- Added Diffie-Hellman key exchange function aesMakeSharedKey()
- Added Visual Studio 2022 project files.
- Added Validate and TestDH VB example programs.