MarshallSoft AES

(Advanced Encryption Standard)

Library for Xbase++

Programmer's Manual

(AES4XB)

Version 5.0

July 17, 2020

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 Xbase++ (AES4XB)** is a toolkit that allows software developers to easily implement strong encryption and decryption into a Windows Xbase+ 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 Xbase** provides information need to compile and run programs in an Xbase programming environment.

The **MarshallSoft Advanced Encryption Standard DLL's** will work under all versions of Windows through Windows 10. A Win32 DLL is provided (a 64-bit DLL is available).

AES4XB includes several Xbase example programs which demonstrate AES encryption and decryption.

The **MarshallSoft Advanced Encryption Standard Library for Xbase** component library supports and has been tested with all versions of 32-bit Alaska Xbase++.

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

For the latest version of the **MarshallSoft AES** software, see www.marshallsoft.com/aes4xb.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 **MarshallSoft Advanced Encryption Library (AES) for Xbase++** component library are as follows:

- Supports both 32-bit and 64-bit Windows.
- Implements the 256-bit "**Advanced Encryption Standard**" (Rijndael)
- Includes cryptographically secure (pseudo) random number generator.
- Supports **ECB** (Electronic Cookbook) mode.
- Supports **CBC** (Cipher Block Chaining) mode.
- Supports SHA-256 cryptographic hash algorithm.
- Supports PKCS7 padding.
- **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 Windows XP through Windows 10.
- Implemented as a **standard** Windows DLL, which will work with all versions of Windows.
- Will run on machines with or without .NET installed.
- Works with all 32-bit versions of Xbase++.
- Does not depend on support libraries. Makes calls to core Windows API functions only.
- Can also be used with any program (in any language) capable of calling Windows API functions such as C/C++, C#, Visual Basic, Delphi, dBase, FoxPro, & 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 dBase example programs with full source code is included. Refer to Section 4 for more details on each of the example programs.

```
1. aesver     Displays AES4XB version
2. TestAES    Performs AES encryption / decryption tests
3. Crypto     Encrypts and/or decrypts a file
4. Encrypt    Encrypts a file
5. Decrypt    Decrypts a file
6. Password   Manages passwords kept encrypted on disk.
7. HashDigest Computes the SHA 256 hash digest.
```

Registration includes one year of free technical support and updates.

1.2 Documentation

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (AES4XB) in the set.

- AES4XB Programmer's Manual `(AES_4XB.PDF)`
- AES User's Manual `(AES_USR.PDF)`
- AES Reference Manual `(AES_REF.PDF)`

The AES_4XB Programmer's Manual (AES_4XB.PDF) is the language specific (Xbase++) manual. All language dependent programming issues such as compiling, compilers and example programs are discussed in this manual. Read this manual first.

The AES User's Manual (AES_USR.PDF) discusses email processing as well as language independent programming issues. Purchasing and license details are also provided.

The AES Reference Manual (AES_REF.PDF) contains details on each individual **AES** function as well as the **AES library** error codes.

The online documentation can be accessed on the **MarshallSoft AES Engine for Xbase++** product page at:

`http://www.marshallsoft.com/aes4xb.htm`

1.3 Example Program

The following example segment demonstrates the use of some of the **MarshallSoft AES for Xbase++** component library functions:

```
Function EncryptFile(KeyBuffer, InFile, OutFile)
Local Code
Local Control
Local Vector
' vector not used in ECM mode
Vector = Chr(0)
Control = "*"
' initialize for encryption in ECB mode
Code = XaesInitAES(@KeyBuffer, @Vector, AES_ECB_MODE, AES_ENCRYPT, @Control)
If Code < 0
  return Code
  exit
endif
' encrypt the file (InFile -> OutFile)
Code = XaesEncryptFile(@Control, @InFile, @OutFile)
return Code
```

1.4 Installation

(1) Before installation of AES4XB, an Xbase++ compiler (any version) should already be installed on your system and tested.

(2) Unzip AES4XB42.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 AES4XB files and copy AES32.DLL to the Windows directory.

The SETUP installation program creates four sub-directories (default \AES4XB) as follows:

```
DOCS - All documentation files
APPS - All example code
DLLS - All DLL’s
```

(4) You are ready to compile and run! For a quick start, load the project file AESVER.PRG.

1.5 Uninstalling

Uninstalling AES4XB is very easy.

First, run UINSTALL.BAT, which will delete AES32.DLL from your Windows directory, typically C:\WINDOWS.

Second, delete the AES4XB project directory created when installing AES4XB.

1.6 Pricing

A developer license for the MarshallSoft AES Library can be registered for $115 USD ($195 with ANSI C source code to the DLL.). Purchasing details can be found in Section 1.4, " How to Purchase", (AES_USR.PDF).

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, the developer will be sent a registered DLL plus a license file (AESxxxxx.LIC, where xxxxx is your Customer ID). 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 (Note that the registered DLL, AES32. DLL does NOT expire). 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 1 and 3 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. Refer to the file UPDATES.TXT located in the /AES4FP/DOCS directory for more information.

2 Library Overview

The **MarshallSoft AES** component library has been tested on multiple computers running Windows XP through Windows 10

2.1 Dynamic Link Libraries

The **MarshallSoft AES** component library includes a Win32 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".

2.2 Keycode

The AES32.DLL has a keycode encoded within it. Your keycode is a 9 or 10-digit decimal number (unless it is 0), and will be found in the file KEYCODE.CH. The keycode for the evaluation version is 0. You will receive a new keycode and a new AES32.DLL after purchasing or updating a developer license. The KEYCODE must be 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 and before installing the registered version, it is best to remove the evaluation version of the AES32.DLL from the Windows search path or delete it.

2.3 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 see the file **aesErrors.txt** for a list of all AES error codes.

2.4 Adding AES4XB to a Project

It is straightforward to add **AES** to both console mode and GUI Xbase++ programs. First, add

```
#INCLUDE "KEYCODE.CH"
#INCLUDE "AES32.CH"
```

after any other `INCLUDE` statements in the Xbase++ program.

Then add

```
nCode = XaesAttach(AES_KEY_CODE, 0)
If nCode < 0 Then
  ? "Cannot attach AES"
  return
endif
```

as the first executed **AES** function.

The keycode (contained in KEYCODE.CH) is 0 for the evaluation version and is a 9-digit number for the purchased version. Rather than include KEYCODE.CH as shown above, the keycode can be pasted directly into the call to **aesAttach**.

Lastly, link your program with AES32.LIB. Refer to the example programs in the AES4XB/APPS subdirectory.

3 Compiler Issues

AES4XB has been compiled and tested with Alaska Xbase++ version v1.3 through Xbase++ v2.0. The SETUP installation program will copy the Lib's and AES32.DLL to the Windows directory. Refer to Section 1.4 "Installation".

3.1 INCLUDE Files

All example programs include two files: `KEYCODE.CH` and `AES32.CH`. The file `AES32.CH` contains all the necessary constants and function declarations for AES4XB, while the file `KEYCODE.CH` contains your keycode (license key), as discussed in Section 2.2.

The Alaska Xbase++ include file DLL.CH is also required. For example,

```
#INCLUDE "DLL.CH"
#INCLUDE "KEYCODE.CH"
#INCLUDE "AES32.CH"
```

The above files can be copied to the Xbase++ INCLUDE directory (where Xbase++ can find it) if so desired.

Note that each function is declared with the prefix character of 'X'

3.2 Compiling and Linking Programs

Details about each of the example programs are provided in Section 4.0 "Example Programs".

To compile and link console mode programs, such as `TestAES.PRG`, use

```
XPP TestAES.PRG
ALINK TestAES.OBJ AES32.LIB
```

To compile and link windows GUI programs, such as `CRYPTO.PRG`, use

```
XPP CRYPTO.PRG
ALINK /SUBSYSTEM:WINDOWS CRYPTO.OBJ AES32.LIB
```

3.3 Xbase++ Compiler

If you don't have the Alaska Software Xbase++ compiler, you can find it on the web at

```
http://www.alaska-software.com
```

4 Example Programs

All example programs are written for 32-bit Xbase. Each has been tested and shows how to correctly use AES functions. It suggested that the developer compile and run the example programs before developing an application using AES4XB.

Note that all AES functions can also be called from all Xbase code modules.

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 is being found and loaded by Windows.

To compile & link:

```
xpp /wl /wu aesver.prg
alink /subsystem:console aesver.obj aes32.lib
```

4.2 TestAES

The **TestAES** example program demonstrates how to encrypt and decrypt messages.

To compile & link:

```
xpp /wl /wu TestAES.prg
alink /subsystem:console TestAES.obj aes32.lib
```

4.3 Crypto

The **Crypto** example (form) program demonstrates how to encrypt a file and to decrypt a (previously encrypted) file.

To compile & link:

```
xpp /wl /wu Crypto.prg
alink /subsystem:windows Crypto.obj aes32.lib
```

4.4 Encrypt

The **Encrypt** example program demonstrates how to encrypt a file.

To compile & link:

```
xpp /wl /wu encrypt.prg
alink /subsystem:console encrypt.obj aes32.lib
```

4.5 Decrypt

The **Decrypt** example program demonstrates how to decrypt a (previously encrypted) file.

To compile & link:

```
xpp /wl /wu decrypt.prg
alink /subsystem:console decrypt.obj aes32.lib
```

4.6 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.

To compile & link:

```
xpp /wl /wu password.prg
alink /subsystem:console password.obj aes32.lib
```

4.7 HashDigest

HashDigest computes the SHA 256 hash digest of a data buffer.

To compile & link:

```
xpp /wl /wu HashDigest.prg
alink /subsystem:console HashDigest.obj aes32.lib
```

5 Revision History

The MarshallSoft AES Engine DLLs (AES32.DLL and AES64.DLL) are written in ANSI C. All programming language versions of AES (C/C++, .NET, Visual Basic, VB .NET, PowerBASIC, Visual FoxPro, Delphi, dBase, Xbase++, COBOL, and Fortran) use the same identical DLLs.

Version 1.0: April 23, 2013.

- Initial release of the Xbase version.

Version 2.0: June 26, 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 PASSWORD example program.

Version 3.0: May 27, 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 29, 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: July 5, 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 10, 2018

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

Version 5.0: July 17, 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.