MarshallSoft AES

(Advanced Encryption Standard)

Library for Delphi

Programmer's Manual

(AES4D)

Version 5.0

July 13, 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 Page 3
  1.1 Features Page 4
  1.2 Documentation Set Page 5
  1.3 Example Program Page 6
  1.4 Installation Page 7
  1.5 Uninstalling Page 7
  1.6 Pricing Page 7
  1.7 Updates Page 7
2 Library Overview Page 8
  2.1 Keycode Page 8
  2.2 Error Display Page 8
  2.3 Adding AES4D to your Project Page 8
  2.4 Passing Pointers Page 9
3 Compiler Issues Page 10
  3.1 Delphi Versions Page 10
  3.2 Delphi Personalities Page 12
  3.3 Compiling Programs Page 12
  3.4 Converting Win32 Projects to Win64 Page 12
4 Example Programs Page 14
  4.1 aesver Page 14
  4.2 TestAES Page 14
  4.3 Crypto Page 14
  4.4 Password Page 14
  4.5 HashDigest Page 15
5 Revision History Page 16

1 Introduction

The **MarshallSoft Advanced Encryption Standard Library** for Delphi **(AES4D)** is a toolkit that allows software developers to easily implement strong encryption and decryption into a Delphi 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 (AES) Programmers Manual for Delphi** provides information need to compile and run programs in a Delphi programming environment.

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

AES4D includes several Delphi example programs which demonstrate AES encryption and decryption.

The **MarshallSoft Advanced Encryption Standard Library** for Delphi component library supports and has been tested with all 32-bit and 64-bit versions of Delphi including:

- Borland Delphi (2.0, 3.0, 4.0, 5.0. 6.0 and 7.0)
- Borland Delphi 8 for .NET
- Borland Delphi 2005 & 2006
- Borland Turbo Delphi
- Codegear Delphi 2007
- Embarcadero Delphi 2009 & 2010
- Embarcadero Delphi XE though XE10

The **MarshallSoft AES** DLLs (AES32.DLL and AES64.DLL) can also be used from any language (C/C++, .NET, C#, Visual Basic, 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/aes4d.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:

- 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 32-bit and 64-bit Windows through Windows 10
- 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 and 64-bit versions of Delphi.
- 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 Delphi example programs with full source code is included. Refer to Section 4 for more details on each of the example programs.

```
1. aesver     Displays AES4D 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. HashDigest Computes the SHA 256 hash digest.
```

1.2 Documentation Set

The complete set of documentation consists of three manuals in Adobe PDF format. This is the first manual (AES_4D) in the set.

- AES4D Programmer's Manual (AES_4D.PDF)
- AES User's Manual (AES_USR.PDF)
- AES Reference Manual (AES_REF.PDF)

The AES_4D Programmer's Manual (AES_4D.PDF) is the language specific (Delphi) 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 licensing details are also provided.

The AES Reference Manual (AES_REF.PDF) contains details on each individual AES function. A list of error codes is also provided.

Online documentation can be accessed on the **MarshallSoft AES Library for Delphi** product page at:

http://www.marshallsoft.com/aes4d.htm

1.3 Example Program

The following example demonstrates the use of some of the **MarshallSoft AES Library for Delphi** component library functions.

```
function EncryptFile(Pass:AnsiString; FileName:AnsiString):Integer;
var
  Code   : Integer;
  CB     : AnsiString;  // control buffer
  Vector : AnsiString;  // initialization vector (not used)
  KeyBuf : AnsiString;  // encryption key
  Source : AnsiString;  // source pathname
  Target : AnsiString;  // target pathname
begin
  {user AES space for control}
  CB := '*';
  {attach AES}
  Code := aesAttach(AES_KEY_CODE, 0);
  if Code < 0 then
  begin
    EncryptFile := -1;
    exit;
  end;
  {create encryption key buffer & initialization vector}
  KeyBuf := StringOfChar(Chr(0), AES_KEY_SIZE);
  Vector := StringOfChar(Chr(0), AES_BLOCK_SIZE);
  {process password phrase}
  Code := aesMakeUserKey(@Pass[1], @KeyBuf[1], 0);
  if Code < 0 then
  begin
    EncryptFile := Code;
    exit
  end;
  {initialize AES for encrypting (ECB mode)}
  Code := aesInitAES(@KeyBuf[1],@Vector[1],AES_ECB_MODE,AES_ENCRYPT,@CB[1]);
  if Code < 0 then
  begin
    {aesInitAES fails}
    EncryptFile := Code;
    exit
  end;
  {construct file names for encryption}
  Source := FileName + Chr(0);
  Target := FileName + '.aes' + Chr(0);
  {encrypt source file to target file}
  Code := aesEncryptFile(@CB[1], @Source[1], @Target[1]);
  if Code < 0 then
  begin
    {aesEncryptFile fails}
    EncryptFile := Code;
    exit
  end;
end;
```

1.4 Installation

(1) Before installation of AES4D, a Delphi compiler (any version) should already be installed.

(2) Unzip AES4D42.ZIP (evaluation version) or AESxxxxx.ZIP (purchased version where xxxxx is your Customer ID) using any Windows unzip program.

(3) Run the installation program, SETUP.EXE, which will install all AES4D files including copying AES32.DLL and AES64.DLL to the Windows directory.

(4) For a quick start, load project file AESVER.DPR

1.5 Uninstalling

Uninstalling AES4D is very easy.

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

Second, delete the **AES** project directory created when installing AES4D.

1.6 Pricing

A developer license for the MarshallSoft AES Library for Delphi can be purchased for $115 USD. Purchasing details can be found in Section 1.4, "How to Purchase", of the AES User's Manual (AES_USR.PDF).

Also see INVOICE.TXT provided with the evaluation version or order directly on our web site at

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

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

1.7 Updates

When a developer license is purchased, the developer will be sent a registered DLL plus a license file (AESxxxx.LIC, where xxxx is your Customer ID). The license file can be used to update the registered DLL's 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 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. Note that the registered DLLs, (AES32. DLL and AES64.DLL) never expire. Refer to the file UPDATES.TXT located in the /AES4D/DOCS directory for more information.

2 Library Overview

The **MarshallSoft AES** component library has been tested on multiple computers running Windows 95/98/Me/XP/2003/2012/Vista/Windows 7 /Windows 8 and Windows NT/2000.

The AES4D library has been tested and works with all versions of Borland (CodeGear) Delphi including Delphi 2 – Delphi 8, Delphi 2005 – Delphi 2010, Embarcadero Delphi XE through XE10 and Turbo Delphi.

The SETUP installation program will copy the AES DLL to the Windows directory and copies the AES4D files to the directory specified (default \AES4D). Four sub-directories are created, as follows:

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

2.1 Keycode

The AES32.DLL has a keycode encoded within it. The keycode is a 9 or 10 digit decimal number and will be found in the file KEYCODE.PAS. The keycode for the evaluation version is 0. The developer will receive a new keycode and a new AES32.DLL after purchasing a license. 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 from the Windows search path or delete it.

2.2 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 **seeErrors.txt** for a list of all Winsock and **AES** error codes.

2.3 Adding AES4D to a Project

Copy AES32.PAS (or AES64.PAS for 64-bit applications) to the same directory as your application program. Also copy the file KEYCODE.PAS to this same directory. You will find these files in the APPS directory (folder) created when you ran SETUP, usually C:\AES4D\APPS.

Next, add a reference to the files copied above to your "uses" clause in your application program. For example,

```
uses aes32, keycode
```

You can leave 'keycode' out above if you put your numerical keycode value (found in `KEYCODE.PAS`) directly into the call to `aesAttach`. For example,

```
{pass the key code}
Code := aesAttach(123456789, 0)
```

Lastly, add a reference to `AES32` to your project file (*.DPR). For example,

```
uses AES32 in 'AES32.PAS'
```

2.4 Passing Pointers

The AES functions accept two kinds of arguments: (1) Integers and (2) Pointers to the first byte of an AnsiChar string.

Integers are passed the same as if calling a Delphi subroutine or function. Pointers are passed as in the following example:

```
var
  Code     : Integer;     {return code}
  PassStr  : AnsiString;  {password phrase string}
  KeyStr   : AnsiString;  {key buffer (always 32 bytes)}
begin
  {create pass phrase string}
  PassStr := 'My secret pass phrase';
  {allocate key buffer string}
  KeyStr := StringOfChar(Chr(0), AES_KEY_SIZE);
  {create key buffer from password phrase}
  Code := aesMakeUserKey(@PassStr[1], @KeyStr[1], 0);
  {KeyStr will now contains the encryption key}
```

In particular, note the syntax for passing ANSI strings PassStr and KeyStr to **aesMakeUserKey** and also that memory is allocated for ANSI string KeyStr before passing to function **aesMakeUserKey**.

Also see the example code segment in section 1.3 above.

3 Compiler Issues

The **MarshallSoft AES Library for Delphi** component library supports all versions of CodeGear (Borland) Delphi for Win32 and Win64 as well as Delphi for .NET as follows:

- Borland Delphi 2, 3, 4, 5, 6, 7 and 8.
- Borland Delphi 2005 (Delphi 9)
- Borland Delphi 2006
- Borland Turbo Delphi
- CodeGear Delphi 2007
- CodeGear Delphi 2009
- CodeGear Delphi 2010
- Embarcadero Delphi XE
- Embarcadero Delphi XE2 (Win32 & Win64)
- . . .
- Embarcadero Delphi XE10 (Win32 & Win64)

3.1 Delphi Versions

Applications written with Delphi link with the same identical DLL as for applications written in all other supported languages, such as C/C++ and Delphi.

3.1.1 Delphi 1

The first release of Borland Delphi (version 1) generated Win16 code.

AES4D does not support 16-bit applications.

3.1.2 Delphi 2

Delphi version 2 and above generates Win32 code and link with AES32DLL. Strings can be much larger than 255 bytes.

Delphi 2 seems to have a problem with some of the string functions. Although the default is "large strings", some of the string functions (such as StrPas) copy only 255 bytes. The MYSTRING.PAS unit contains a replacement unit to use instead of StrPCopy.

3.1.3 Delphi 3

Delphi 3 also has some problems with PChar string functions such as StrPCopy. See the previous section.

3.1.4 Delphi 4, 5, and 6.

There are no known Delphi problems impacting our example programs in Delphi version 4 and above.

3.1.5 Delphi 7 and 8.

Beginning in Delphi 7, the filename of a unit must match the unit name. Delphi 8 was a .NET only release.

3.1.6 Delphi 2005, Delphi 2006, and Delphi 2007

The Delphi 2005/2006/2007 compilers support both Win32 and the Microsoft .NET Framework.

When loading Delphi for Win32 projects with Delphi 2005/2006/2007, a window entitled "Project Upgrade" will be displayed:

This project must be upgraded before it can be opened. Please select which project type you wish to target:

```
( ) Delphi for .NET
( ) Delphi for Win32
```

Choose "Delphi for Win32" for all projects except "*.bdsproj" projects, which are Delphi for .NET projects.

3.1.7 Delphi 2009, Delphi 2010 and Delphi XE

In Delphi 2009 the definition of PChar was changed from a pointer to an 8-bit character to a pointer to a 16-bit character, also known as a “wide character”. For this reason PAnsiChar must be used rather than PChar for pointers to buffers that are passed to AES functions. Refer to the Win32 example programs in the APPS directory.

3.1.8 Delphi XE2 Through XE10

Delphi XE2 through XE8 can create both 32-bit and 64-bit executables. Note that although a 32-bit executable can run on both 32-bit and 64-bit Windows machines, a 64-bit executable can run only on a 64-bit Windows machine.

The example programs are configured for Win32. The file **x64.zip** contains the example programs after modification to X64.

3.2 Delphi Personalities

Beginning with Delphi 2003, Delphi has two “personalities”: (1) Win32 Delphi and (2) Delphi for .NET. Win32 Delphi programs are a continuation of the Delphi language as seen in earlier versions of Delphi. Delphi .NET is a version of Delphi designed to use the Microsoft .NET Framework.

Starter editions of (Embarcadero) Delphi support the creation of 32-bit applications but not 64-bit applications, even if running on a 64-bit computer. That is, starter editions support the “Win32 personality” but not the “Win64 personality”.

3.3 Compiling Programs

The example programs are compiled from the Delphi development environment using the provided Delphi project files (*.DPR).

Refer to Section 4.0 "Example Programs" for more details on each of the example programs.

3.4 Converting Win32 projects to Win64

Note that 64-bit executables can be created with Delphi XE2. Although a 32-bit executable can run on both 32-bit and 64-bit Windows machines, a 64-bit executable can run only on a 64-bit Windows machine.

Recall that starter editions of Embarcadero Delphi support the Win32 personality but not the Win64 personality.

The file **x64.zip** contains the example programs after modification to Win64 (x64). The process to convert Win32 programs to Win64 programs manually is as follows:

Edit Files

1. Open file *_pgm.pas using any text editor.
2. Replace aes32 with aes64 in the "uses" clause.

You should end up with a line in *_pgm.pas that looks like:

```
Display, aes64, KeyCode,
```

3. Open file *_prj.dpr using any text editor.
4. Replace aes32 with aes64.

You should end up with a line in *_prj.dpr that looks like:

```
aes64 in 'aes64.pas',
```

Delete Win32 Files

1. Delete all *.dcu files that may have been created under Win32 Delphi.
2. Delete all *.dproj files that may have been created under Win32 Delphi.

Change Win32 to Win64 in Delphi XE2 Project Manager

Start Delphi XE2, load the project *_prj.dpr, then open the Delphi Project Manager.

1. Right click on "Target Platforms (Win32)" and click on "Add Platform...".
2. When the "Select Platform" window is displayed, click "64-bit Windows".
3. Click [+] box to left of "Target Platforms (Win32)"
4. Right click on "32-bit Windows", then click "Remove Platform"

Convert Referenced Files

You must also convert display.pas by replacing all reference to aes32 with aes64.

The file **x64.zip** contains the example programs after modification to X64.

4 Example Programs

Several Delphi example programs are included in **MarshallSoft AES Library for Delphi.** Before writing your own programs, compile and run the example programs.

Each of the following example programs uses the "display" unit and the "AES" unit:

```
DISPLAY.PAS   : Display unit source code.
AES.PAS       : AES Unit source code.
```

The DISPLAY.PAS unit is used to display text in Delphi memos. DISPLAY.PAS contains 4 procedures:

```
DisplayChar   : Displays character.
DisplayString : Displays string.
DisplayLine   : Displays line.
DisplayError  : Displays error message.
```

4.1 aesVer (AES Version)

The AESVER example program displays the **AES** library version number and registration string and verifies that AES32.DLL or AES64.DLL is being found and loaded by Windows.

The **aesver** project files are:

```
VER_PRJ.DPR  : Delphi project file.
VER_PGM.DFM  : Delphi form file.
VER_PGM.PAS  : Program source code.
```

4.2 TestAES

TestAES demonstrates how to encrypt and decrypt messages.

The **TestAES** project files are:

```
TestAES_PRJ.DPR  : Delphi project file.
TestAES_PGM.DFM  : Delphi form file.
TestAES_PGM.PAS  : Program source code.
```

4.3 Crypto

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

The **Crypto** project files are:

```
Crypto_PRJ.DPR  : Delphi project file.
Crypto_PGM.DFM  : Delphi form file.
Crypto_PGM.PAS  : Program source code.
```

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.

The **Password** project files are:

```
Password_PRJ.DPR  : Delphi project file.
Password_PGM.DFM  : Delphi form file.
Password_PGM.PAS  : Program source code.
```

4.5 HashDigest

HashDigest computes the SHA 256 hash digest of a salted password.

The **HashDigest** project files are:

```
HashDigest_PRJ.DPR  : Delphi project file.
HashDigest_PGM.DFM  : Delphi form file.
HashDigest_PGM.PAS  : Program source code.
```

5 Revision History

The MarshallSoft AES Library DLL (AES32.DLL and AES64.DLL) is written in ANSI C. All programming language versions of AES (C/C++, Delphi, Visual Basic, PowerBASIC, Visual FoxPro, Delphi, dBase, Xbase++, COBOL, and FORTRAN) use the same AES32.DLL or AES64.DLL.

Version 1.0: May 7, 2013

- Initial Delphi release.

Version 2.0: June 18, 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 support for Embarcadero Delphi XE5 and XE6.
- Added PASSWORD example program.

Version 3.0: May 29, 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.
- Added support for Embarcadero Delphi XE7 and Delphi XE8.

Version 4.0: December 6, 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 26, 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 11, 2018

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

Version 5.0: July 13, 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.