```
        {
            string original = "Here is some data to encrypt!";

            // Create a new instance of the Aes
            // class.  This generates a new key and initialization
            // vector (IV).
            using (Aes myAes = Aes.Create())
            {

                // Encrypt the string to an array of bytes.
                byte[] encrypted = EncryptStringToBytes_Aes(original, myAes.Key,
myAes.IV);

                // Decrypt the bytes to a string.
                string roundtrip = DecryptStringFromBytes_Aes(encrypted,
myAes.Key, myAes.IV);

                //Display the original data and the decrypted data.
                Console.WriteLine("Original:   {0}", original);
                Console.WriteLine("Round Trip: {0}", roundtrip);
            }
        }
        static byte[] EncryptStringToBytes_Aes(string plainText, byte[] Key,
byte[] IV)
        {
            // Check arguments.
            if (plainText == null || plainText.Length <= 0)
                throw new ArgumentNullException("plainText");
            if (Key == null || Key.Length <= 0)
                throw new ArgumentNullException("Key");
            if (IV == null || IV.Length <= 0)
                throw new ArgumentNullException("IV");
            byte[] encrypted;

            // Create an Aes object
            // with the specified key and IV.
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Key;
                aesAlg.IV = IV;


                // Create an encryptor to perform the stream transform.
                ICryptoTransform encryptor = aesAlg.CreateEncryptor(aesAlg.Key,
aesAlg.IV);

                // Create the streams used for encryption.
                using (MemoryStream msEncrypt = new MemoryStream())
                {
                    using (CryptoStream csEncrypt = new CryptoStream(msEncrypt,
encryptor, CryptoStreamMode.Write))
                    {
                        using (StreamWriter swEncrypt = new
StreamWriter(csEncrypt))
                        {
                            //Write all data to the stream.
                            swEncrypt.Write(plainText);
                            swEncrypt.Write(plainText);
                        }
                        encrypted = msEncrypt.ToArray();
                    }
                }
            }

            // Return the encrypted bytes from the memory stream.
            return encrypted;
        }

        static string DecryptStringFromBytes_Aes(byte[] cipherText, byte[] Key,
byte[] IV)
        {
            // Check arguments.
            if (cipherText == null || cipherText.Length <= 0)
                throw new ArgumentNullException("cipherText");
            if (Key == null || Key.Length <= 0)
                throw new ArgumentNullException("Key");
            if (IV == null || IV.Length <= 0)
                throw new ArgumentNullException("IV");

            // Declare the string used to hold
            // the decrypted text.
            string plaintext = null;

            // Create an Aes object
            // with the specified key and IV.
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Key;
                aesAlg.IV = IV;

                // Create a decryptor to perform the stream transform.
                ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key,
aesAlg.IV);

                // Create the streams used for decryption.
                using (MemoryStream msDecrypt = new MemoryStream(cipherText))
                {
                    using (CryptoStream csDecrypt = new CryptoStream(msDecrypt,
decryptor, CryptoStreamMode.Read))
                    {
                        using (StreamReader srDecrypt = new
StreamReader(csDecrypt))
                        {

                            // Read the decrypted bytes from the decrypting
stream
                            // and place them in a string.
                            plaintext = srDecrypt.ReadToEnd();
                        }
                    }
                }
            }
            return plaintext;
        }
    }
}
```

Constructors

Aes()	Initializes a new instance of the Aes class.

Fields

BlockSizeValue	Represents the block size, in bits, of the cryptographic operation. (Inherited from SymmetricAlgorithm)
FeedbackSizeValue	Represents the feedback size, in bits, of the cryptographic operation. (Inherited from SymmetricAlgorithm)
IVValue	Represents the initialization vector (IV) for the symmetric algorithm. (Inherited from SymmetricAlgorithm)
KeySizeValue	Represents the size, in bits, of the secret key used by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
KeyValue	Represents the secret key for the symmetric algorithm. (Inherited from SymmetricAlgorithm)
LegalBlockSizesValue	Specifies the block sizes, in bits, that are supported by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
LegalKeySizesValue	Specifies the key sizes, in bits, that are supported by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
ModeValue	Represents the cipher mode used in the symmetric algorithm. (Inherited from SymmetricAlgorithm)
PaddingValue	Represents the padding mode used in the symmetric algorithm. (Inherited from SymmetricAlgorithm)

Properties

BlockSize	Gets or sets the block size, in bits, of the cryptographic operation. (Inherited from SymmetricAlgorithm)

	(Inherited from SymmetricAlgorithm)
FeedbackSize	Gets or sets the feedback size, in bits, of the cryptographic operation for the Cipher Feedback (CFB) and Output Feedback (OFB) cipher modes. (Inherited from SymmetricAlgorithm)
IV	Gets or sets the initialization vector (IV) for the symmetric algorithm. (Inherited from SymmetricAlgorithm)
Key	Gets or sets the secret key for the symmetric algorithm. (Inherited from SymmetricAlgorithm)
KeySize	Gets or sets the size, in bits, of the secret key used by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
LegalBlockSizes	Gets the block sizes, in bits, that are supported by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
LegalKeySizes	Gets the key sizes, in bits, that are supported by the symmetric algorithm. (Inherited from SymmetricAlgorithm)
Mode	Gets or sets the mode for operation of the symmetric algorithm. (Inherited from SymmetricAlgorithm)
Padding	Gets or sets the padding mode used in the symmetric algorithm. (Inherited from SymmetricAlgorithm)

Methods

Clear()	Releases all resources used by the SymmetricAlgorithm class. (Inherited from SymmetricAlgorithm)
Create()	Creates a cryptographic object that is used to perform the symmetric algorithm.
Create(String)	Creates a cryptographic object that specifies the implementation of AES to use to perform the symmetric algorithm.
CreateDecryptor()	Creates a symmetric decryptor object with the current Key property and initialization vector (IV). (Inherited from SymmetricAlgorithm)
CreateDecryptor(Byte[], Byte[])	When overridden in a derived class, creates a symmetric decryptor object with the specified Key property and initialization vector (IV). (Inherited from SymmetricAlgorithm)
CreateEncryptor()	Creates a symmetric encryptor object with the current Key property and initialization vector (IV). (Inherited from SymmetricAlgorithm)

CreateEncryptor(Byte[], Byte[])	When overridden in a derived class, creates a symmetric encryptor object with the specified Key property and initialization vector (IV). (Inherited from SymmetricAlgorithm)
DecryptCbc(Byte[], Byte[], PaddingMode)	Decrypts data using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
DecryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, PaddingMode)	Decrypts data using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
DecryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode)	Decrypts data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
DecryptCfb(Byte[], Byte[], PaddingMode, Int32)	Decrypts data using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
DecryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, PaddingMode, Int32)	Decrypts data using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
DecryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	Decrypts data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
DecryptEcb(Byte[], PaddingMode)	Decrypts data using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
DecryptEcb(ReadOnlySpan<Byte>, PaddingMode)	Decrypts data using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
DecryptEcb(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode)	Decrypts data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
Dispose()	Releases all resources used by the current instance of the SymmetricAlgorithm class. (Inherited from SymmetricAlgorithm)
Dispose(Boolean)	Releases the unmanaged resources used by the SymmetricAlgorithm and optionally releases the managed resources. (Inherited from SymmetricAlgorithm)
EncryptCbc(Byte[], Byte[], PaddingMode)	Encrypts data using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
EncryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, PaddingMode)	Encrypts data using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)

EncryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode)	Encrypts data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
EncryptCfb(Byte[], Byte[], PaddingMode, Int32)	Encrypts data using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
EncryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, PaddingMode, Int32)	Encrypts data using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
EncryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	Encrypts data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
EncryptEcb(Byte[], PaddingMode)	Encrypts data using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
EncryptEcb(ReadOnlySpan<Byte>, PaddingMode)	Encrypts data using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
EncryptEcb(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode)	Encrypts data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
Equals(Object)	Determines whether the specified object is equal to the current object. (Inherited from Object)
GenerateIV()	When overridden in a derived class, generates a random initialization vector (IV) to use for the algorithm. (Inherited from SymmetricAlgorithm)
GenerateKey()	When overridden in a derived class, generates a random key (Key) to use for the algorithm. (Inherited from SymmetricAlgorithm)
GetCiphertextLengthCbc(Int32, PaddingMode)	Gets the length of a ciphertext with a given padding mode and plaintext length in CBC mode. (Inherited from SymmetricAlgorithm)
GetCiphertextLengthCfb(Int32, PaddingMode, Int32)	Gets the length of a ciphertext with a given padding mode and plaintext length in CFB mode. (Inherited from SymmetricAlgorithm)
GetCiphertextLengthEcb(Int32, PaddingMode)	Gets the length of a ciphertext with a given padding mode and plaintext length in ECB mode. (Inherited from SymmetricAlgorithm)
GetHashCode()	Serves as the default hash function. (Inherited from Object)

GetType()	Gets the Type of the current instance. (Inherited from Object)
MemberwiseClone()	Creates a shallow copy of the current Object. (Inherited from Object)
ToString()	Returns a string that represents the current object. (Inherited from Object)
TryDecryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, Int32, PaddingMode)	Attempts to decrypt data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryDecryptCbcCore(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	When overridden in a derived class, attempts to decrypt data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryDecryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, Int32, PaddingMode, Int32)	Attempts to decrypt data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
TryDecryptCfbCore(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32, Int32)	When overridden in a derived class, attempts to decrypt data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
TryDecryptEcb(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	Attempts to decrypt data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryDecryptEcbCore(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	When overridden in a derived class, attempts to decrypt data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryEncryptCbc(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, Int32, PaddingMode)	Attempts to encrypt data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryEncryptCbcCore(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	When overridden in a derived class, attempts to encrypt data into the specified buffer, using CBC mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryEncryptCfb(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, Int32, PaddingMode, Int32)	Attempts to encrypt data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)

TryEncryptCfbCore(ReadOnlySpan<Byte>, ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32, Int32)	When overridden in a derived class, attempts to encrypt data into the specified buffer, using CFB mode with the specified padding mode and feedback size. (Inherited from SymmetricAlgorithm)
TryEncryptEcb(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	Attempts to encrypt data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
TryEncryptEcbCore(ReadOnlySpan<Byte>, Span<Byte>, PaddingMode, Int32)	When overridden in a derived class, attempts to encrypt data into the specified buffer, using ECB mode with the specified padding mode. (Inherited from SymmetricAlgorithm)
ValidKeySize(Int32)	Determines whether the specified key size is valid for the current algorithm. (Inherited from SymmetricAlgorithm)

Applies to

Product	Versions
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7 Preview 6
.NET Framework	3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8
.NET Standard	1.3, 1.4, 1.6, 2.0, 2.1
Xamarin.iOS	10.8
Xamarin.Mac	3.0

Recommended content

RijndaelManaged Class (System.Security.Cryptography)

Accesses the managed version of the Rijndael algorithm. This class cannot be inherited.

AesManaged Class (System.Security.Cryptography)

Provides a managed implementation of the Advanced Encryption Standard (AES) symmetric algorithm.

Rijndael Class (System.Security.Cryptography)

Represents the base class from which all implementations of the Rijndael symmetric encryption algorithm must inherit.

SymmetricAlgorithm.CreateEncryptor Method (System.Security.Cryptography)

Creates a symmetric encryptor object.

AesCryptoServiceProvider Class (System.Security.Cryptography)

Performs symmetric encryption and decryption using the Cryptographic Application Programming Interfaces (CAPI) implementation of the Advanced Encryption Standard (AES) algorithm.

Decrypting data

Learn how to decrypt data in .NET, using a symmetric algorithm or an asymmetric algorithm.

Encrypting data

Learn how to encrypt data in .NET, using a symmetric algorithm or an asymmetric algorithm.

Aes.Create Method (System.Security.Cryptography)

Creates a cryptographic object that is used to perform the symmetric algorithm.

Show more