System.Text.UTF8Encoding Class

```
[ILAsm]
.class public serializable UTF8Encoding extends System.Text.Encoding

[C#]
public class UTF8Encoding: Encoding
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
    - CLSCompliantAttribute(true)

Summary

Represents a UTF8 character `System.Text.Encoding`.

Inherits From: System.Text.Encoding

Library: BCL

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

`System.Text.UTF8Encoding` encodes Unicode characters using the UTF-8 encoding (UCS Transformation Format, 8-bit form). This encoding supports all Unicode character values.

[*Note:* UTF-8 encodes Unicode characters with a variable number of bytes per character. This encoding is optimized for the lower 127 ASCII characters, yielding an efficient mechanism to encode English in an internationalizable way. The UTF-8 identifier is the Unicode byte order mark (0xFEFF) written in UTF-8 (0xEF 0xBB 0xBF). The byte order mark is used to distinguish UTF-8 text from other encodings.

This class offers an error-checking feature that can be turned on when an instance of the class is constructed. Certain methods in this class check for invalid sequences of surrogate pairs. If error-checking is turned on and an invalid sequence is detected, `System.ArgumentException` is thrown. If error-checking is not turned on and an invalid sequence is detected, no exception is thrown and execution continues in a method-defined manner. For more information regarding surrogate pairs, see `System.Globalization.UnicodeCategory`.

]

UTF8Encoding(System.Boolean, System.Boolean) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(bool
encoderShouldEmitUTF8Identifier, bool throwOnInvalidBytes)


[C#]
public UTF8Encoding(bool encoderShouldEmitUTF8Identifier, bool
throwOnInvalidBytes)
```

Summary

Constructs a new instance of the `System.Text.UTF8Encoding` class using the specified `System.Boolean` flags.

Parameters

Parameter	Description
encoderShouldEmitUTF8Identifier	A `System.Boolean` that indicates whether the Unicode byte order mark in UTF-8 is recognized or emitted when reading from or writing to a `System.IO.Stream`.
throwOnInvalidBytes	A `System.Boolean` that indicates whether error-checking is turned on for the current instance.

UTF8Encoding(System.Boolean) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(bool
encoderShouldEmitUTF8Identifier)

[C#]
public UTF8Encoding(bool encoderShouldEmitUTF8Identifier)
```

Summary

Constructs a new instance of the `System.Text.UTF8Encoding` class with the specified `System.Boolean` that indicates whether the Unicode byte order mark in UTF-8 is recognized or emitted when reading from or writing to a `System.IO.Stream`.

Parameters

Parameter	Description
encoderShouldEmitUTF8Identifier	A `System.Boolean` that indicates whether the Unicode byte order mark in UTF-8 is recognized or emitted when reading from or writing to a `System.IO.Stream`.

Description

This constructor is equivalent to `System.Text.UTF8Encoding` (*encoderShouldEmitUTF8Identifier*, `false`).

[*Note:* By default, this constructor turns error-checking off for the new instance.]

UTF8Encoding() Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor()


[C#]
public UTF8Encoding()
```

Summary

Constructs a new instance of the `System.Text.UTF8Encoding` class.

Description

This constructor is equivalent to `System.Text.UTF8Encoding` (`false, false`).

[*Note:* By default, this constructor turns error-checking off for the new instance.]

UTF8Encoding.Equals(System.Object) Method

```
[ILAsm]
.method public hidebysig virtual bool Equals(object value)


[C#]
public override bool Equals(object value)
```

Summary

Determines whether the current instance and the specified `System.Object` represent the same type and value.

Parameters

Parameter	Description
value	A `System.Object` to compare with the current instance.

Return Value

`true` if value is a `System.Text.UTF8Encoding` and represents the same type and value as the current instance; otherwise, `false`.

Description

[*Note:* This method overrides `System.Object.Equals`.]

UTF8Encoding.GetByteCount(System.String) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetByteCount(string chars)

[C#]
public override int GetByteCount(string chars)
```

Summary

Determines the number of bytes required to encode the characters in the specified `System.String` as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
chars	A `System.String` to encode as a `System.Text.UTF8Encoding`.

Return Value

A `System.Int32` that specifies the number of bytes necessary to encode *chars* as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid characters are ignored and do not affect the return value, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetByteCount`.]

Exceptions

Exception	Condition
System.ArgumentNullException	*chars* is `null`.
System.ArgumentException	Error-checking is turned on for the current instance and *chars* contains an invalid surrogate sequence.

System.ArgumentOutOfRangeException	The return value is greater than `System.Int32.MaxValue`.

UTF8Encoding.GetByteCount(System.Char [], System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetByteCount(class
System.Char[] chars, int32 index, int32 count)


[C#]
public override int GetByteCount(char[] chars, int index, int count)
```

Summary

Determines the number of bytes required to encode the specified range of characters in the specified Unicode character array as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
chars	The `System.Char` array to encode as a `System.Text.UTF8Encoding`.
index	A `System.Int32` that specifies the first index of *chars* to encode.
count	A `System.Int32` that specifies the number of characters to encode.

Return Value

A `System.Int32` containing the number of bytes necessary to encode the range in *chars* from *index* to *index + count* - 1 as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid characters are ignored and do not affect the return value, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoder.GetByteCount`.

]

Exceptions

Exception	Condition
System.ArgumentNullException	*chars* is `null`.

System.ArgumentOutOfRangeException	The return value is greater than `System.Int32.MaxValue`. -or- *index* or *count* is less than zero. -or- *index* and *count* do not specify a valid range in *chars* (i.e. (*index* + *count*) > *chars*.Length).
System.ArgumentException	Error-checking is turned on for the current instance and *chars* contains an invalid surrogate sequence.

UTF8Encoding.GetBytes(System.String, System.Int32, System.Int32, System.Byte[], System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetBytes(string s, int32
charIndex, int32 charCount, class System.Byte[] bytes, int32
byteIndex)


[C#]
public override int GetBytes(string s, int charIndex, int charCount,
byte[] bytes, int byteIndex)
```

Summary

Encodes the specified range of the specified `System.String` into the specified range of the specified `System.Byte` array as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
s	The `System.String` to encode as a `System.Text.UTF8Encoding`.
charIndex	A `System.Int32` that specifies the first index of *s* to encode.
charCount	A `System.Int32` that specifies the number of characters to encode.
bytes	The `System.Byte` array to encode into.
byteIndex	A `System.Int32` that specifies the first index of *bytes* to encode into.

Return Value

A `System.Int32` that indicates the number of bytes encoded into *bytes* as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid characters are ignored and are not encoded into *bytes*, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetBytes`.]

Exceptions

Exception	Condition
System.ArgumentException	*bytes* does not contain sufficient space to store the encoded characters. -or- Error-checking is turned on for the current instance and *chars* contains an invalid surrogate sequence.
System.ArgumentNullException	*chars* or *bytes* is `null`.
System.ArgumentOutOfRangeException	*charIndex*, *charCount*, or *byteIndex* is less than zero. -or- (*s*.Length - *charIndex*) < *charCount*. -or- *byteIndex* >= *bytes*.Length.

UTF8Encoding.GetBytes(System.String) Method

```
[ILAsm]
.method public hidebysig virtual class System.Byte[] GetBytes(string
s)

[C#]
public override byte[] GetBytes(string s)
```

Summary

Encodes the specified `System.String` as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
s	The `System.String` to encode as a `System.Text.UTF8Encoding`.

Return Value

A `System.Byte` array containing the values encoded from *s* as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid characters are ignored and are not encoded into the returned `System.Byte` array, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetBytes`.]

Exceptions

Exception	Condition
System.ArgumentException	Error-checking is turned on for the current instance and *s* contains an invalid surrogate sequence.
System.ArgumentNullException	*s* is `null`.

UTF8Encoding.GetBytes(System.Char[], System.Int32, System.Int32, System.Byte[], System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetBytes(class System.Char[]
chars, int32 charIndex, int32 charCount, class System.Byte[] bytes,
int32 byteIndex)

[C#]
public override int GetBytes(char[] chars, int charIndex, int
charCount, byte[] bytes, int byteIndex)
```

Summary

Encodes the specified range of the specified `System.Char` array into the specified range of the specified `System.Byte` array as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
chars	The `System.Char` array to encode as a `System.Text.UTF8Encoding`.
charIndex	A `System.Int32` that specifies the first index of *chars* to encode.
charCount	A `System.Int32` that specifies the number of characters to encode.
bytes	The `System.Byte` array to encode into.
byteIndex	A `System.Int32` that specifies the first index of *bytes* to encode into.

Return Value

A `System.Int32` that indicates the number of bytes encoded into *bytes* as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid characters are ignored and are not encoded into *bytes*, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetBytes`.

`System.Text.UTF8Encoding.GetByteCount` can be used to determine the exact number of bytes that will be produced for a given range of characters. Alternatively, `System.Text.UTF8Encoding.GetMaxByteCount` can be used to determine the maximum number of bytes that will be produced for a specified

number of characters, regardless of the actual character values.

]

Exceptions

Exception	Condition
System.ArgumentException	*bytes* does not contain sufficient space to store the encoded characters. -or- Error-checking is turned on for the current instance and *chars* contains an invalid surrogate sequence.
System.ArgumentNullException	*chars* or *bytes* is `null`.
System.ArgumentOutOfRangeException	*charIndex*, *charCount*, or *byteIndex* is less than zero. -or- (*chars*.Length - *charIndex*) < *charCount*. -or- *byteIndex* > *bytes*.Length.

UTF8Encoding.GetCharCount(System.Byte[], System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetCharCount(class
System.Byte[] bytes, int32 index, int32 count)


[C#]
public override int GetCharCount(byte[] bytes, int index, int count)
```

Summary

Returns the number of characters produced by decoding the specified range of the specified `System.Byte` array as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
bytes	The `System.Byte` array to decode as a `System.Text.UTF8Encoding`.
index	A `System.Int32` that specifies the first index of *bytes* to decode.
count	A `System.Int32` that specifies the number of bytes to decode.

Return Value

A `System.Int32` that indicates the number of characters produced by decoding the range in *bytes* from *index* to *index* + *count* - 1 as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid bytes are ignored and do not affect the return value, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetCharCount`.]

Exceptions

Exception	Condition
System.ArgumentNullException	*bytes* is `null`.

System.ArgumentOutOfRangeException	*index* or *count* is less than zero. -or- *index* and *count* do not specify a valid range in *bytes* (i.e. (*index* + *count*) > *bytes*.Length).
System.ArgumentException	Error-checking is turned on for the current instance and *bytes* contains an invalid surrogate sequence.

UTF8Encoding.GetChars(System.Byte[], System.Int32, System.Int32, System.Char[], System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetChars(class System.Byte[]
bytes, int32 byteIndex, int32 byteCount, class System.Char[] chars,
int32 charIndex)

[C#]
public override int GetChars(byte[] bytes, int byteIndex, int
byteCount, char[] chars, int charIndex)
```

Summary

Decodes the specified range of the specified `System.Byte` array into the specified range of the specified `System.Char` array as a `System.Text.UTF8Encoding`.

Parameters

Parameter	Description
bytes	The `System.Byte` array to decode as a `System.Text.UTF8Encoding`.
byteIndex	A `System.Int32` that specifies the first index of *bytes* to decode.
byteCount	A `System.Int32` that specifies the number of bytes to decode.
chars	The `System.Char` array to decode into.
charIndex	A `System.Int32` that specifies the first index of *chars* to decode into.

Return Value

The number of characters decoded into *chars* as a `System.Text.UTF8Encoding`.

Description

If error-checking is turned off and an invalid surrogate sequence is detected, the invalid bytes are ignored and are not encoded into *chars*, and no exception is thrown.

[*Note:* This method overrides `System.Text.Encoding.GetChars`.

`System.Text.UTF8Encoding.GetCharCount` can be used to determine the exact number of characters that will be produced for a specified range of bytes. Alternatively, `System.Text.UTF8Encoding.GetMaxCharCount` can be used to determine the maximum number of characters that will be produced for a specified number of bytes, regardless of the actual byte values.

]

Exceptions

Exception	Condition
System.ArgumentException	*chars* does not contain sufficient space to store the decoded characters. -or- Error-checking is turned on for the current instance and *bytes* contains an invalid surrogate sequence.
System.ArgumentNullException	*bytes* or *chars* is `null`.
System.ArgumentOutOfRangeException	*byteIndex*, *byteCount*, or *charIndex* is less than zero. -or- (*bytes*.Length - *byteIndex*) < *byteCount*. -or- *charIndex* > *chars*.Length.

UTF8Encoding.GetDecoder() Method

```
[ILAsm]
.method public hidebysig virtual class System.Text.Decoder
GetDecoder()


[C#]
public override Decoder GetDecoder()
```

Summary

Returns a `System.Text.Decoder` for the current instance.

Return Value

A `System.Text.Decoder` for the current instance.

Description

[*Note:* This method overrides `System.Text.Encoding.GetDecoder`.

Contrary to `System.Text.UTF8Encoding.GetChars`, a decoder can convert partial sequences of bytes into partial sequences of characters by maintaining the appropriate state between the conversions.

]

UTF8Encoding.GetEncoder() Method

```
[ILAsm]
.method public hidebysig virtual class System.Text.Encoder
GetEncoder()

[C#]
public override Encoder GetEncoder()
```

Summary

Returns a `System.Text.Encoder` for the current instance.

Return Value

A `System.Text.Encoder` for the current instance.

Description

[*Note:* This method overrides `System.Text.Encoding.GetEncoder`.

Contrary to `System.Text.UTF8Encoding.GetBytes`, an encoder can convert partial sequences of characters into partial sequences of bytes by maintaining the appropriate state between the conversions.

]

UTF8Encoding.GetHashCode() Method

```
[ILAsm]
.method public hidebysig virtual int32 GetHashCode()

[C#]
public override int GetHashCode()
```

Summary

Generates a hash code for the current instance.

Return Value

A `System.Int32` value containing a hash code for the current instance

Description

The algorithm used to generate the hash code is unspecified.

[*Note:* This method overrides `System.Object.GetHashCode`.]

UTF8Encoding.GetMaxByteCount(System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetMaxByteCount(int32
charCount)

[C#]
public override int GetMaxByteCount(int charCount)
```

Summary

Returns the maximum number of bytes required to encode the specified number of characters as a `System.Text.UTF8Encoding`, regardless of the actual character values.

Parameters

Parameter	Description
charCount	A `System.Int32` that specifies the number of characters to encode as a `System.Text.UTF8Encoding`.

Return Value

A `System.Int32` that specifies the maximum number of bytes required to encode *charCount* characters as a `System.Text.UTF8Encoding`.

Description

[*Note:* This method overrides `System.Text.Encoding.GetMaxByteCount`.

This method can be used to determine an appropriate buffer size for byte arrays passed to `System.Text.UTF8Encoding.GetBytes`. Using this minimum buffer size can help ensure that no buffer overflow exceptions will occur.

]

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*charCount* < 0.

UTF8Encoding.GetMaxCharCount(System. Int32) Method

```
[ILAsm]
.method public hidebysig virtual int32 GetMaxCharCount(int32
byteCount)

[C#]
public override int GetMaxCharCount(int byteCount)
```

Summary

Returns the maximum number of characters produced by decoding the specified number of bytes as a `System.Text.UTF8Encoding`, regardless of the actual byte values.

Parameters

Parameter	Description
byteCount	A `System.Int32` that specifies the number of bytes to decode as a `System.Text.UTF8Encoding`.

Return Value

A `System.Int32` that specifies the maximum number of characters produced by decoding *byteCount* bytes as a `System.Text.UTF8Encoding`.

Description

[*Note:* This method overrides `System.Text.Encoding.GetMaxCharCount`.

This method can be used to determine an appropriate minimum buffer size for character arrays passed to `System.Text.UTF8Encoding.GetChars`. Using this minimum buffer size can help ensure that no buffer overflow exceptions will occur.

]

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*byteCount* < 0.

UTF8Encoding.GetPreamble() Method

```
[ILAsm]
.method public hidebysig virtual class System.Byte[] GetPreamble()


[C#]
public override byte[] GetPreamble()
```

Summary

Returns the bytes used at the beginning of a stream to determine which encoding a file was created with.

Return Value

A `System.Byte` array containing the UTF-8 encoding preamble.

Description

[*Note:* This method overrides `System.Text.Encoding.GetPreamble`.

`System.Text.UTF8Encoding.GetPreamble` returns the Unicode byte order mark (U+FEFF) written in UTF-8 (0xef, 0xbb, 0xbf) if this instance was constructed with a request to emit the UTF-8 identifier.

]