

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]
- *Version:* 1.0.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

1 information regarding surrogate pairs, see
2 **System.Globalization.UnicodeCategory.**]

3

1 UTF8Encoding() Constructor

```
2 [ILASM]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public UTF8Encoding()
```

6 Summary

7 Constructs a new instance of the **System.Text.UTF8Encoding** class.

8 Description

9 This constructor is equivalent to **System.Text.UTF8Encoding (false,**
10 **false)**.

11
12 [*Note:* By default, this constructor turns error-checking off for the new
13 instance.]

14

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
<i>encoderShouldEmitUTF8Identifier</i>	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.]

1 UTF8Encoding(System.Boolean, 2 System.Boolean) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(bool  
5 encoderShouldEmitUTF8Identifier, bool throwOnInvalidBytes)  
  
6 [C#]  
7 public UTF8Encoding(bool encoderShouldEmitUTF8Identifier,  
8 bool throwOnInvalidBytes)
```

9 Summary

10 Constructs a new instance of the **System.Text.UTF8Encoding** class
11 using the specified **System.Boolean** flags.

12 Parameters

13
14

Parameter	Description
<i>encoderShouldEmitUTF8Identifier</i>	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 .
<i>throwOnInvalidBytes</i>	A System.Boolean that indicates whether error-checking is turned on for the current instance.

15
16
17

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
<i>value</i>	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.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
<i>chars</i>	The System.Char array to encode as a System.Text.UTF8Encoding .
<i>index</i>	A System.Int32 that specifies the first index of <i>chars</i> to encode.
<i>count</i>	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* 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	<i>chars</i> is null .
System.ArgumentOutOfRangeException	The return value is greater than System.Int32.MaxValue .

1
2
3

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

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
<i>chars</i>	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	<i>chars</i> is null .
System.ArgumentException	Error-checking is turned on for the current instance and <i>chars</i> contains an invalid surrogate sequence.

1
2
3

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

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
<i>s</i>	The System.String to encode as a System.Text.UTF8Encoding .
<i>charIndex</i>	A System.Int32 that specifies the first index of <i>s</i> to encode.
<i>charCount</i>	A System.Int32 that specifies the number of characters to encode.
<i>bytes</i>	The System.Byte array to encode into.
<i>byteIndex</i>	A System.Int32 that specifies the first index of <i>bytes</i> 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	<p><i>bytes</i> does not contain sufficient space to store the encoded characters.</p> <p>-or-</p> <p>Error-checking is turned on for the current instance and <i>chars</i> contains an invalid surrogate sequence.</p>
System.ArgumentNullException	<i>chars</i> or <i>bytes</i> is null .
System.ArgumentOutOfRangeException	<p><i>charIndex</i>, <i>charCount</i>, or <i>byteIndex</i> is less than zero.</p> <p>-or-</p> <p>$(s.Length - charIndex) < charCount$.</p> <p>-or-</p> <p>$byteIndex > bytes.Length$.</p>

1
2
3

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
<i>chars</i>	The System.Char array to encode as a System.Text.UTF8Encoding .
<i>charIndex</i>	A System.Int32 that specifies the first index of <i>chars</i> to encode.
<i>charCount</i>	A System.Int32 that specifies the number of characters to encode.
<i>bytes</i>	The System.Byte array to encode into.
<i>byteIndex</i>	A System.Int32 that specifies the first index of <i>bytes</i> 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,

1 **System.Text.UTF8Encoding.GetMaxByteCount** can be used to
2 determine the maximum number of bytes that will be produced for a
3 specified number of characters, regardless of the actual character
4 values.]

5 **Exceptions**

6
7

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

8
9
10

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
<i>bytes</i>	The System.Byte array to decode as a System.Text.UTF8Encoding .
<i>index</i>	A System.Int32 that specifies the first index of <i>bytes</i> to decode.
<i>count</i>	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* 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	<i>bytes</i> is null .

1
2
3

System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero. -or- <i>index</i> and <i>count</i> do not specify a valid range in <i>bytes</i> (i.e. (<i>index</i> + <i>count</i>) > <i>bytes.Length</i>).
System.ArgumentException	Error-checking is turned on for the current instance and <i>bytes</i> 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
<i>bytes</i>	The System.Byte array to decode as a System.Text.UTF8Encoding .
<i>byteIndex</i>	A System.Int32 that specifies the first index of <i>bytes</i> to decode.
<i>byteCount</i>	A System.Int32 that specifies the number of bytes to decode.
<i>chars</i>	The System.Char array to decode into.
<i>charIndex</i>	A System.Int32 that specifies the first index of <i>chars</i> 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,

1 **System.Text.UTF8Encoding.GetMaxCharCount** can be used to
2 determine the maximum number of characters that will be produced
3 for a specified number of bytes, regardless of the actual byte values.]

4 **Exceptions**

5
6

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

7
8
9

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.]

1 UTF8Encoding.GetEncoder() Method

```
2 [ILASM]  
3 .method public hidebysig virtual class System.Text.Encoder  
4 GetEncoder()  
  
5 [C#]  
6 public override Encoder GetEncoder()
```

7 Summary

8 Returns a **System.Text.Encoder** for the current instance.

9 Return Value

10

11 A **System.Text.Encoder** for the current instance.

12 Description

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

14

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

18

1 UTF8Encoding.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4  
5 [C#]  
6 public override int GetHashCode()
```

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

9

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

11 Description

12 The algorithm used to generate the hash code is unspecified.

13

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

15

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
<i>charCount</i>	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	<i>charCount</i> < 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
<i>byteCount</i>	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	<i>byteCount</i> < 0.

1
2
3

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.]