

1 System.Text.UTF8Encoding Class

```
2 [ILAsm]  
3 .class public serializable UTF8Encoding extends System.Text.Encoding  
4 [C#]  
5 public class UTF8Encoding: Encoding
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

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

14 Inherits From: `System.Text.Encoding`

15

16 **Library:** BCL

17

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

20

21 Description

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

25

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

31

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

39

40]

41

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, bool  
8 throwOnInvalidBytes)
```

9 Summary

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

12 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.
<i>throwOnInvalidBytes</i>	A System.Boolean that indicates whether error-checking is turned on for the current instance.

13

14

1 UTF8Encoding(System.Boolean) Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(bool  
4 encoderShouldEmitUTF8Identifier)  
  
5 [C#]  
6 public UTF8Encoding(bool encoderShouldEmitUTF8Identifier)
```

7 Summary

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

11 Parameters

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

12

13 Description

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

16
17 [*Note:* By default, this constructor turns error-checking off for the new instance.]
18
19

20

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, false)`.

10
11 [*Note:* By default, this constructor turns error-checking off for the new instance.]
12
13

14

1 UTF8Encoding.Equals(System.Object) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual bool Equals(object value)  
4 [C#]  
5 public override bool Equals(object value)
```

6 Summary

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

9 Parameters

Parameter	Description
<i>value</i>	A <code>System.Object</code> to compare with the current instance.

10

11 Return Value

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

14 Description

15 [*Note:* This method overrides `System.Object.Equals`.]
16
17

18

1 UTF8Encoding.GetByteCount(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual int32 GetByteCount(string chars)  
5 [C#]  
6 public override int GetByteCount(string chars)
```

7 Summary

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

10 Parameters

Parameter	Description
<i>chars</i>	A System.String to encode as a System.Text.UTF8Encoding.

11 Return Value

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

15 Description

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

18 [Note: This method overrides System.Text.Encoding.GetByteCount.]
19
20
21

22 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.
System.ArgumentOutOfRangeException	The return value is greater than

	System.Int32.MaxValue.
--	------------------------

1

2

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* - 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	<i>chars</i> is null.

<p>System.ArgumentOutOfRangeException</p>	<p>The return value is greater than <code>System.Int32.MaxValue</code>.</p> <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. $(index + count) > chars.Length$).</p>
<p>System.ArgumentException</p>	<p>Error-checking is turned on for the current instance and <i>chars</i> contains an invalid surrogate sequence.</p>

1

2

1 UTF8Encoding.GetBytes(System.String, 2 System.Int32, System.Int32, System.Byte[], 3 System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual int32 GetBytes(string s, int32 charIndex,  
6 int32 charCount, class System.Byte[] bytes, int32 byteIndex)  
  
7 [C#]  
8 public override int GetBytes(string s, int charIndex, int charCount,  
9 byte[] bytes, int byteIndex)
```

10 Summary

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

13 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.

14 15 Return Value

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

18 Description

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

21
22 [Note: This method overrides System.Text.Encoding.GetBytes.]
23
24

25 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>$byteIndex \geq bytes.Length$.</p>

1

2

1 UTF8Encoding.GetBytes(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual class System.Byte[] GetBytes(string s)  
5 [C#]  
6 public override byte[] GetBytes(string s)
```

7 Summary

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

9 Parameters

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

10

11 Return Value

12 A System.Byte array containing the values encoded from s as a
13 System.Text.UTF8Encoding.

14 Description

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

18

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

20

21

22 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.

23

24

1 UTF8Encoding.GetBytes(System.Char[], 2 System.Int32, System.Int32, System.Byte[], 3 System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual int32 GetBytes(class System.Char[] chars,  
6 int32 charIndex, int32 charCount, class System.Byte[] bytes, int32  
7 byteIndex)  
  
8 [C#]  
9 public override int GetBytes(char[] chars, int charIndex, int charCount,  
10 byte[] bytes, int byteIndex)
```

11 Summary

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

14 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.

15 16 Return Value

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

19 Description

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

22
23 *[Note:* This method overrides System.Text.Encoding.GetBytes.

24
25 System.Text.UTF8Encoding.GetByteCount can be used to determine the exact number
26 of bytes that will be produced for a given range of characters. Alternatively,

1 `System.Text.UTF8Encoding.GetMaxByteCount` can be used to determine the maximum
 2 number of bytes that will be produced for a specified number of characters, regardless
 3 of the actual character values.
 4
 5]

6 **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>$(chars.Length - charIndex) < charCount$.</p> <p>-or-</p> <p>$byteIndex > bytes.Length$.</p>

7

8

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

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.

1

2

1 UTF8Encoding.GetChars(System.Byte[], 2 System.Int32, System.Int32, System.Char[], 3 System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual int32 GetChars(class System.Byte[] bytes,  
6 int32 byteIndex, int32 byteCount, class System.Char[] chars, int32  
7 charIndex)  
  
8 [C#]  
9 public override int GetChars(byte[] bytes, int byteIndex, int byteCount,  
10 char[] chars, int charIndex)
```

11 Summary

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

14 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.

15 16 Return Value

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

18 Description

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

21 [Note: This method overrides System.Text.Encoding.GetChars.

22
23
24 System.Text.UTF8Encoding.GetCharCount can be used to determine the exact number
25 of characters that will be produced for a specified range of bytes. Alternatively,
26 System.Text.UTF8Encoding.GetMaxCharCount can be used to determine the maximum

1 number of characters that will be produced for a specified number of bytes, regardless
 2 of the actual byte values.
 3
 4]

5 **Exceptions**

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

6

7

1 UTF8Encoding.GetDecoder() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual class System.Text.Decoder GetDecoder()  
4 [C#]  
5 public override Decoder GetDecoder()
```

6 Summary

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

8 Return Value

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

10 Description

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

12

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

16

17]

18

1 UTF8Encoding.GetEncoder() Method

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

6 Summary

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

8 Return Value

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

10 Description

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

12

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

16

17]

18

1 UTF8Encoding.GetHashCode() Method

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

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

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

10 Description

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

12

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

14

15

16

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 <code>System.Int32</code> that specifies the number of characters to encode as a <code>System.Text.UTF8Encoding</code> .

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
<code>System.ArgumentOutOfRangeException</code>	<i>charCount</i> < 0.

1
2 **UTF8Encoding.GetMaxCharCount(System.Int32)**
3 **Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual int32 GetMaxCharCount(int32 byteCount)  
6 [C#]  
7 public override int GetMaxCharCount(int byteCount)
```

8 **Summary**

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

11 **Parameters**

Parameter	Description
<i>byteCount</i>	A <code>System.Int32</code> that specifies the number of bytes to decode as a <code>System.Text.UTF8Encoding</code> .

12
13 **Return Value**

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

16 **Description**

17 [Note: This method overrides `System.Text.Encoding.GetMaxCharCount`.
18
19 This method can be used to determine an appropriate minimum buffer size for character
20 arrays passed to `System.Text.UTF8Encoding.GetChars`. Using this minimum buffer size
21 can help ensure that no buffer overflow exceptions will occur.
22
23]

24 **Exceptions**

Exception	Condition
<code>System.ArgumentOutOfRangeException</code>	<i>byteCount</i> < 0.

25
26

1 UTF8Encoding.GetPreamble() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual class System.Byte[] GetPreamble()  
4 [C#]  
5 public override byte[] GetPreamble()
```

6 Summary

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

9 Return Value

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

11 Description

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

14 System.Text.UTF8Encoding.GetPreamble returns the Unicode byte order mark
15 (U+FEFF) written in UTF-8 (0xef, 0xbb, 0xbf) if this instance was constructed with a
16 request to emit the UTF-8 identifier.
17

18]
19