

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

1 UTF8Encoding(System.Boolean)

2 Constructor

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

8 Summary

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

13 Parameters

14
15

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 .

16

17 Description

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

20

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

23

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

1 UTF8Encoding.Equals(System.Object)

2 Method

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

7 Summary

8 Determines whether the current instance and the specified
9 **System.Object** represent the same type and value.

10 Parameters

11
12

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

13
14
15

14 Return Value

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

18 Description

19 [Note: This method overrides **System.Object.Equals**.]
20

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 .

	<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.

1
2
3

1 UTF8Encoding.GetByteCount(System.String) Method

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

8 Summary

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

11 Parameters

12
13

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

14
15
16

Return Value

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

19 Description

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

23
24
25

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

26 Exceptions

27
28

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
<p>System.ArgumentException</p>	<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>
<p>System.ArgumentNullException</p>	<p><i>chars</i> or <i>bytes</i> is null.</p>
<p>System.ArgumentOutOfRangeException</p>	<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

1 UTF8Encoding.GetBytes(System.String)

2 Method

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

8 Summary

9 Encodes the specified **System.String** as a
10 **System.Text.UTF8Encoding**.

11 Parameters

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

15 Return Value

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

19 Description

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

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

25 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- $(chars.Length - charIndex) < charCount$. -or- $byteIndex > bytes.Length$.

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 .

<p>System.ArgumentOutOfRangeException</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>bytes</i> (i.e. (<i>index</i> + <i>count</i>) > <i>bytes.Length</i>).</p>
<p>System.ArgumentException</p>	<p>Error-checking is turned on for the current instance and <i>bytes</i> contains an invalid surrogate sequence.</p>

1
2
3

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

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

11 Summary

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

15 Parameters

16
17

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.

18
19
20

19 Return Value

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

23 Description

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

27
28
29

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

30 **System.Text.UTF8Encoding.GetCharCount** can be used to
31 determine the exact number of characters that will be produced for a
32 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- $(bytes.Length - byteIndex) < byteCount$. -or- $charIndex > chars.Length$.

7
8
9

1 UTF8Encoding.GetDecoder() Method

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

7 Summary

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

9 Return Value

10

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

12 Description

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

14

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

18

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 [C#]  
5 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

1 UTF8Encoding.GetMaxByteCount(System.I 2 nt32) Method

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

8 Summary

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

12 Parameters

13
14

Parameter	Description
<i>charCount</i>	A System.Int32 that specifies the number of characters to encode as a System.Text.UTF8Encoding .

15
16
17

16 Return Value

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

21 Description

22 [Note: This method overrides
23 **System.Text.Encoding.GetMaxByteCount**.
24

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

29 Exceptions

30
31

Exception	Condition
System.ArgumentOutOfRangeException	<i>charCount</i> < 0.

32
33
34

1 UTF8Encoding.GetMaxCharCount(System. 2 Int32) Method

```
3 [ILASM]  
4 .method public hidebysig virtual int32  
5 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
10 specified number of bytes as a **System.Text.UTF8Encoding**,
11 regardless of the actual byte values.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

21 Description

22 [Note: This method overrides
23 **System.Text.Encoding.GetMaxCharCount**.
24

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

29 Exceptions

30
31

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

1
2
3

1 UTF8Encoding.GetPreamble() Method

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

7 Summary

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

10 Return Value

11

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

13 Description

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

15

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

19