

# 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
<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> .
<i>throwOnInvalidBytes</i>	A <code>System.Boolean</code> 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
<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> .

## 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
<i>value</i>	A <code>System.Object</code> 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
<i>chars</i>	A <code>System.String</code> to encode as a <code>System.Text.UTF8Encoding</code> .

## 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
<b>System.ArgumentNullException</b>	<i>chars</i> is null.
<b>System.ArgumentException</b>	Error-checking is turned on for the current instance and <i>chars</i> contains an invalid surrogate sequence.
<b>System.ArgumentOutOfRangeException</b>	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 <code>System.Char</code> array to encode as a <code>System.Text.UTF8Encoding</code> .
<i>index</i>	A <code>System.Int32</code> that specifies the first index of <i>chars</i> to encode.
<i>count</i>	A <code>System.Int32</code> 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
<b>System.ArgumentNullException</b>	<i>chars</i> is null.



<b>System.ArgumentOutOfRangeException</b>	<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. <math>(index + count) &gt; chars.Length</math>).</p>
<b>System.ArgumentException</b>	<p>Error-checking is turned on for the current instance and <i>chars</i> contains an invalid surrogate sequence.</p>

1

2

# 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 <code>System.String</code> to encode as a <code>System.Text.UTF8Encoding</code> .
<i>charIndex</i>	A <code>System.Int32</code> that specifies the first index of <i>s</i> to encode.
<i>charCount</i>	A <code>System.Int32</code> that specifies the number of characters to encode.
<i>bytes</i>	The <code>System.Byte</code> array to encode into.
<i>byteIndex</i>	A <code>System.Int32</code> 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
<b>System.ArgumentException</b>	<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>
<b>System.ArgumentNullException</b>	<i>chars</i> or <i>bytes</i> is null.
<b>System.ArgumentOutOfRangeException</b>	<p><i>charIndex</i>, <i>charCount</i>, or <i>byteIndex</i> is less than zero.</p> <p>-or-</p> <p><math>(s.Length - charIndex) &lt; charCount</math>.</p> <p>-or-</p> <p><math>byteIndex \geq bytes.Length</math>.</p>

1

2

# 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
<code>s</code>	The <code>System.String</code> to encode as a <code>System.Text.UTF8Encoding</code> .

### 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
<b>System.ArgumentException</b>	Error-checking is turned on for the current instance and <code>s</code> contains an invalid surrogate sequence.
<b>System.ArgumentNullException</b>	<code>s</code> 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 <code>System.Char</code> array to encode as a <code>System.Text.UTF8Encoding</code> .
<i>charIndex</i>	A <code>System.Int32</code> that specifies the first index of <i>chars</i> to encode.
<i>charCount</i>	A <code>System.Int32</code> that specifies the number of characters to encode.
<i>bytes</i>	The <code>System.Byte</code> array to encode into.
<i>byteIndex</i>	A <code>System.Int32</code> 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 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
<b>System.ArgumentException</b>	<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.
<b>System.ArgumentNullException</b>	<i>chars</i> or <i>bytes</i> is null.
<b>System.ArgumentOutOfRangeException</b>	<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> .

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
<b>System.ArgumentNullException</b>	<i>bytes</i> is null.

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

1

2



# 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 <code>System.Byte</code> array to decode as a <code>System.Text.UTF8Encoding</code> .
<i>byteIndex</i>	A <code>System.Int32</code> that specifies the first index of <i>bytes</i> to decode.
<i>byteCount</i>	A <code>System.Int32</code> that specifies the number of bytes to decode.
<i>chars</i>	The <code>System.Char</code> array to decode into.
<i>charIndex</i>	A <code>System.Int32</code> 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, `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
<b>System.ArgumentException</b>	<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.
<b>System.ArgumentNullException</b>	<i>bytes</i> or <i>chars</i> is null.
<b>System.ArgumentOutOfRangeException</b>	<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> .

6

7

# 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
<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.

# 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 <code>System.Int32</code> that specifies the number of bytes to decode as a <code>System.Text.UTF8Encoding</code> .

## 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
<code>System.ArgumentOutOfRangeException</code>	<i>byteCount</i> < 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.

]