System.Text.UnicodeEncoding Class

```
[ILASM]
.class public serializable UnicodeEncoding extends
System.Text.Encoding

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

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Summary

Represents a Unicode implementation of **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.UnicodeEncoding encodes each Unicode character in UTF-16, i.e. as two consecutive bytes. Both little-endian and big-endian encodings are supported.

[*Note:* On little-endian platforms such as Intel machines, it is generally more efficient to store Unicode characters in little-endian. However, many other platforms can store Unicode characters in big-endian. Unicode files can be distinguished by the presence of the byte order mark (U+FEFF), which is written as either 0xfe 0xff or 0xff 0xfe.

This **System.Text.Encoding** implementation can detect a byte order mark automatically and switch byte orders, based on a parameter specified in the constructor.

ISO/IEC 10646 defines UCS-2 and UCS-4. UCS-4 is a four-byte (32-bit) encoding containing 2^{31} code positions, divided into 128 groups of 256 planes. Each plane contains 2^{16} code positions. UCS-2 is a two-byte (16-bit) encoding containing the 2^{16} code positions of UCS-4 for

which the upper two bytes are zero, known as Plane Zero or the Basic Multilingual Plane (BMP). For example, the code position for LATIN CAPITAL LETTER A in UCS-4 is 0x00000041 whereas in UCS-2 it is 0x0041.

ISO/IEC 10646 also defines UTF-16, which stands for "UCS Transformation Format for 16 Planes of Group 00". UTF-16 is a two byte encoding that uses an extension mechanism to represent 2^{21} code positions. UTF-16 represents code positions in Plane Zero by its UCS-2 code value and code positions in Planes 1 through 16 by a pair of special code values, called surrogates. UTF-16 is equivalent to the Unicode Standard. For a detailed description of UTF-16 and surrogates, see "The Unicode Standard Version 3.0" Appendix C.]

UnicodeEncoding() Constructor

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

[C#]
public UnicodeEncoding()
```

Summary

Constructs and initializes a new instance of the **System.Text.UnicodeEncoding** class.

Description

The new instance uses little-endian encoding and includes the Unicode byte-order mark in conversions.

UnicodeEncoding(System.Boolean, System.Boolean) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(bool
bigEndian, bool byteOrderMark)

[C#]
public UnicodeEncoding(bool bigEndian, bool byteOrderMark)
```

Summary

Constructs and initializes a new instance of the **System.Text.UnicodeEncoding** class using the specified Boolean flags.

Parameters

Parameter	Description
bigEndian	A **System.Boolean** value that specifies the byte-ordering to use for the new instance. Specify **true** to use big-endian ordering; specify **false** to use little-endian ordering.
byteOrderMark	A **System.Boolean** value that specifies whether to include the Unicode byte order mark in translated strings. Specify **true** to include the Unicode byte-order mark; otherwise, specify **false**.

UnicodeEncoding.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	The **System.Object** to compare to the current instance.

Return Value

true if *value* represents the same type and value as the current instance. If *value* is a null reference or is not an instance of **System.Text.UnicodeEncoding**, returns **false**.

Description

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

UnicodeEncoding.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 exact number of bytes required to encode the specified range of the specified array of characters as Unicode-encoded characters.

Parameters

Parameter	Description
chars	A **System.Char** array to encode as Unicode-encoded characters.
index	A **System.Int32** that specifies the first index of *chars* to encode.
count	A **System.Int32** that specifies the number of elements in *chars* to encode.

Return Value

A **System.Int32** whose value equals the number of bytes required to encode the range in *chars* from *index* to *index* + *count* as Unicode-encoded characters.

Description

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

Exceptions

Exception	Condition
System.ArgumentNullException	*chars* is **null**.
System.ArgumentOutOfRangeException	*index* < 0.

	-or- *count* < 0. -or- *index* and *count* do not specify a valid range in *chars* (i.e. (*index* + *count*) > *chars*.Length).

1
2
3

UnicodeEncoding.GetByteCount(System.String) Method

```
[ILASM]
.method public hidebysig virtual int32 GetByteCount(string
s)

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

Summary

Returns the number of bytes required to encode the specified string as Unicode-encoded characters.

Parameters

Parameter	Description
s	A **System.String** to encode as Unicode-encoded characters.

Return Value

A **System.Int32** containing the number of bytes needed to encode *s* as Unicode-encoded characters.

Description

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

Exceptions

Exception	Condition
System.ArgumentNullException	*s* is **null**.

UnicodeEncoding.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 string into the specified range of the specified array of bytes as Unicode-encoded characters.

Parameters

Parameter	Description
s	A **System.String** to encode as Unicode-encoded characters.
charIndex	A **System.Int32** that specifies the first index of *s* from which to encode.
charCount	A **System.Int32** that specifies the number of elements in *s* to encode.
bytes	A **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** whose value equals the number of bytes encoded into *bytes* as Unicode-encoded characters.

Description

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

Exceptions

Exception	Condition
System.ArgumentException	*bytes* does not contain sufficient space to store the encoded characters.

System.ArgumentNullException	*s* is **null**. -or- *bytes* is **null**.
System.ArgumentOutOfRangeException	*charIndex* < 0. -or- *charCount* < 0. -or- *byteIndex* < 0. -or- *charIndex* and *charCount* do not specify a valid range in *s* (i.e. (*charIndex* + *charCount*) > *s*.Length). -or- *byteIndex* > *bytes*.Length.

1
2
3

UnicodeEncoding.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 string as Unicode-encoded characters.

Parameters

Parameter	Description
s	A **System.String** to encode as Unicode-encoded characters.

Return Value

A **System.Byte** array containing the encoded representation of *s* as Unicode-encoded characters.

Description

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

Exceptions

Exception	Condition
System.ArgumentNullException	*s* is **null**.

UnicodeEncoding.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 character array into the specified range of the specified byte array as Unicode-encoded characters.

Parameters

Parameter	Description
chars	A **System.Char** array of characters to encode as Unicode-encoded characters.
charIndex	A **System.Int32** that specifies the first index of *chars* to encode.
charCount	A **System.Int32** that specifies the number of elements in *chars* to encode.
bytes	A **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** containing the number of bytes encoded into *bytes* as Unicode-encoded characters.

Description

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

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

determine the maximum number of bytes that will be produced for a given 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.
System.ArgumentNullException	*chars* is **null**. -or- *bytes* is **null**.
System.ArgumentOutOfRangeException	*charIndex* < 0. -or- *charCount* < 0. -or- *byteIndex* < 0. -or- *charIndex* and *charCount* do not specify a valid range in *chars* (i.e. (*charIndex* + *charCount)* > *chars*.Length). -or- *byteIndex* > *bytes*.Length.

UnicodeEncoding.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

Determines the exact number of characters that will be produced by decoding the specified range of the specified array of bytes as Unicode-encoded characters.

Parameters

Parameter	Description
bytes	A **System.Byte** array to decode as Unicode-encoded characters.
index	A **System.Int32** that specifies the first index in *bytes* to decode.
count	A **System.Int32** that specifies the number of elements in *bytes* to decode.

Return Value

A **System.Int32** whose value equals the number of characters a call to **System.Text.UnicodeEncoding.GetChars** will produce if presented with the specified range of *bytes* as Unicode-encoded characters.

Description

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

Exceptions

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

System.ArgumentOutOfRangeException	*index* < 0. -or- *count* < 0. -or- *index* and *count* do not specify a valid range in *bytes* (i.e. (*index* + *count*) > *bytes*.Length).

1
2
3

UnicodeEncoding.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 array of bytes into the specified range of the specified array of characters as Unicode-encoded characters.

Parameters

Parameter	Description
bytes	A **System.Byte** array to decode as Unicode-encoded characters.
byteIndex	A **System.Int32** that specifies the first index of *bytes* from which to decode.
byteCount	A **System.Int32** that specifies the number of elements in *bytes* to decode.
chars	A **System.Char** array to decode into.
charIndex	A **System.Int32** that specifies the first index of *chars* to store the decoded bytes.

Return Value

A **System.Int32** containing the number of characters decoded into *chars* as Unicode-encoded characters.

Description

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

System.Text.UnicodeEncoding.GetCharCount can be used to determine the exact number of characters that will be produced for a given range of bytes. Alternatively, **System.Text.UnicodeEncoding.GetMaxCharCount** can be used to

determine the maximum number of characters that will be produced for a given 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.
System.ArgumentNullException	*chars* is **null**. -or- *bytes* is **null**.
System.ArgumentOutOfRangeException	*byteIndex* < 0. -or- *byteCount* < 0. -or- *charIndex* < 0. -or- *byteIndex* and *byteCount* do not specify a valid range in *bytes* (i.e. (*byteIndex* + *byteCount)* > *bytes*.Length). -or- *charIndex* > *chars*.Length.

UnicodeEncoding.GetDecoder() Method

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

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

Summary

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

Return Value

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

Description

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

Unlike the **System.Text.UnicodeEncoding.GetChars** method, the **System.Text.Decoder.GetChars** method provided by a **System.Text.Decoder** object can convert partial sequences of bytes into partial sequences of characters by maintaining the appropriate state between the conversions.

This implementation returns a decoder that simply forwards calls to **System.Text.UnicodeEncoding.GetCharCount** and **System.Text.UnicodeEncoding.GetChars** to the corresponding methods of the current instance. It is recommended that encoding implementations that requires state to be maintained between successive conversions override this method and return an instance of an appropriate decoder implementation.]

UnicodeEncoding.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** containing the hash code for the current instance.

Description

The algorithm used to generate the hash code is unspecified.

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

UnicodeEncoding.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 Unicode-encoded characters, regardless of the actual character values.

Parameters

Parameter	Description
charCount	A **System.Int32** whose value represents a number of characters to encode as Unicode-encoded characters.

Return Value

A **System.Int32** containing the maximum number of bytes required to encode *charCount* characters as Unicode-encoded characters.

Description

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

Use this method to determine an appropriate minimum buffer size for byte arrays passed to **System.Text.UnicodeEncoding.GetBytes** or **System.Text.Encoder.GetBytes** for the current instance. Using this minimum buffer size can help ensure that buffer overflow exceptions do not occur.]

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*charCount* < 0.

UnicodeEncoding.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 Unicode-encoded characters, regardless of the actual byte values.

Parameters

Parameter	Description
byteCount	A **System.Int32** specifies the number of bytes to decode as Unicode-encoded characters.

Return Value

A **System.Int32** containing the maximum number of characters that would be produced by decoding *byteCount* bytes as Unicode-encoded characters.

Description

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

Use this method to determine an appropriate minimum buffer size for byte arrays passed to **System.Text.UnicodeEncoding.GetChars** or **System.Text.Encoding.GetChars** for the current instance. Using this minimum buffer size can help ensure that no buffer overflow exceptions will occur.]

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*byteCount* < 0.

1
2
3

UnicodeEncoding.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 **System.IO.Stream** instance to determine which **System.Text.Encoding** implementation the stream was created with.

Return Value

A **System.Byte** array that identifies the **System.Text.Encoding** implementation used to create a **System.IO.Stream**.

Description

System.Text.UnicodeEncoding.GetPreamble returns the Unicode byte order mark (U+FEFF) in either big-endian or little-endian order, according the ordering that the current instance was initialized with.

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