

System.UInt16 Structure

```
[ILAsm]
.class public sequential sealed serializable UInt16 extends
System.ValueType implements System.IComparable, System.IFormattable,
System.IComparable`1<unsigned int16>, System.IEquatable`1<unsigned int16>

[C#]
public struct UInt16: IComparable, IFormattable, IComparable<UInt16>,
IEquatable<UInt16>
```

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)

Type Attributes:

- CLSCompliantAttribute(false)

Implements:

- **System.IComparable**
- **System.IFormattable**
- **System.IComparable<System.UInt16>**
- **System.IEquatable<System.UInt16>**

Summary

Represents a 16-bit unsigned integer.

Inherits From: System.ValueType

Library: BCL

Thread Safety: This type is safe for multithreaded operations.

Description

The `System.UInt16` data type represents integer values ranging from 0 to positive 65,535 (hexadecimal 0xFFFF).

UInt16.MaxValue Field

```
[ILAsm]  
.field public static literal unsigned int16 MaxValue = 65535  
  
[C#]  
public const ushort MaxValue = 65535
```

Summary

Contains the maximum value for the `System.UInt16` type.

Description

The value of this constant is 65,535 (hexadecimal 0xFFFF).

1 UInt16.MinValue Field

2 [ILAsm]

3 .field public static literal unsigned int16 MinValue = 0

4 [C#]

5 public const ushort MinValue = 0

6 Summary

7 Contains the minimum value for the `System.UInt16` type.

8 Description

9 The value of this constant is 0.

UInt16.CompareTo(System.Object) Method

```
[ILAsm]  
.method public final hidebysig virtual int32 CompareTo(object value)  
  
[C#]  
public int CompareTo(object value)
```

Summary

Returns the sort order of the current instance compared to the specified `System.Object`.

Parameters

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

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of the current instance as compared to *value*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Return Value	Description
A negative number	Current instance < <i>value</i> .
Zero	Current instance == <i>value</i> .
A positive number	Current instance > <i>value</i> , or <i>value</i> is a null reference.

Description

[*Note:* This method is implemented to support the `System.IComparable` interface.]

1 **Exceptions**

2

3

Exception	Condition
System.ArgumentException	<i>value</i> is not a <i>System.UInt16</i> and is not a null reference.

4

5

6

UInt16.CompareTo(System.UInt16) Method

```
[ILAsm]  
.method public final hidebysig virtual int32 CompareTo(unsigned int16  
value)  
  
[C#]  
public int CompareTo(ushort value)
```

Summary

Returns the sort order of the current instance compared to the specified `System.UInt16`.

Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt16</code> to compare to the current instance.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of the current instance as compared to *value*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Return Value	Description
A negative number	Current instance < <i>value</i> .
Zero	Current instance == <i>value</i> .
A positive number	Current instance > <i>value</i> .

Description

[*Note:* This method is implemented to support the `System.IComparable<UInt16>` interface.]

UInt16.Equals(System.Object) Method

```
[ILAsm]  
.method public hidebysig virtual bool Equals(object obj)  
  
[C#]  
public override bool Equals(object obj)
```

Summary

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

Parameters

Parameter	Description
<i>obj</i>	The <code>System.Object</code> to compare to the current instance.

Return Value

`true` if *obj* represents the same value and type as the current instance. If *obj* is a null reference or is not an instance of `System.UInt16`, returns `false`.

Description

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

UInt16.Equals(System.UInt16) Method

```
[ILAsm]  
.method public hidebysig virtual bool Equals(unsigned int16 obj)  
  
[C#]  
public override bool Equals(ushort obj)
```

Summary

Determines whether the current instance and the specified `System.UInt16` represent the same type.

Parameters

Parameter	Description
<i>obj</i>	The <code>System.UInt16</code> to compare to the current instance.

Return Value

`true` if *obj* represents the same value and type as the current instance; otherwise, `false`.

Description

[*Note:* This method is implemented to support the `System.IEquatable<UInt16>` interface.]

UInt16.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()`.]

1 UInt16.Parse(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 Parse(string s)  
4 [C#]  
5 public static ushort Parse(string s)
```

6 Summary

7 Returns the specified System.String converted to a System.UInt16 value.

8 Type Attributes:

- 9 · CLSCompliantAttribute(false)

10 Parameters

Parameter	Description
s	A System.String containing the value to convert. The string is interpreted using the System.Globalization.NumberStyles.Integer style.

14 Return Value

16 The System.UInt16 value obtained from s.

17 Description

18 This version of System.UInt16.Parse is equivalent to System.UInt16.Parse(s,
19 System.Globalization.NumberStyles.Integer, null).

20 The string s is parsed using the formatting information in a
22 System.Globalization.NumberFormatInfo initialized for the current system culture.

23 [Note: For more information, see
24 System.Globalization.NumberFormatInfo.CurrentInfo.]

28 This method is not CLS-compliant. For a CLS-compliant alternative use
29 System.Int32.Parse(System.String).

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct style.
System.OverflowException	s represents a number greater than <code>System.UInt16.MaxValue</code> or less than <code>System.UInt16.MinValue</code> .

Example

This example demonstrates parsing a string to a `System.UInt16`.

[C#]

```
using System;
public class UInt16ParseClass {
    public static void Main() {
        string str = " 100 ";
        Console.WriteLine("String: \"{0}\" <UInt16> {1}", str, UInt16.Parse(str));
    }
}
```

The output is

```
String: " 100 " <UInt16> 100
```

UInt16.Parse(System.String, System.Globalization.NumberStyles) Method

```
[ILAsm]  
.method public hidebysig static unsigned int16 Parse(string s, valuetype  
System.Globalization.NumberStyles style)  
  
[C#]  
public static ushort Parse(string s, NumberStyles style)
```

Summary

Returns the specified `System.String` converted to a `System.UInt16` value.

Type Attributes:

- `CLSCompliantAttribute(false)`

Parameters

Parameter	Description
<i>s</i>	A <code>System.String</code> containing the value to convert. The string is interpreted using the style specified by <i>style</i> .
<i>style</i>	Zero or more <code>System.Globalization.NumberStyles</code> values that specify the style of <i>s</i> . Specify multiple values for <i>style</i> using the bitwise OR operator. If <i>style</i> is a null reference, the string is interpreted using the <code>System.Globalization.NumberStyles.Integer</code> style.

Return Value

The `System.UInt16` value obtained from *s*.

Description

This version of `System.UInt16.Parse` is equivalent to `System.UInt16.Parse(s, style, null)`.

The string *s* is parsed using the formatting information in a `System.Globalization.NumberFormatInfo` initialized for the current system culture.
[Note: For more information, see

1 `System.Globalization.NumberFormatInfo.CurrentInfo.]`

2
3
4
5 This method is not CLS-compliant. For a CLS-compliant alternative use
6 `System.Int32.Parse(System.String,System.Globalization.NumberStyles).`

7 **Exceptions**

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct style.
System.OverflowException	s represents a number greater than System.UInt16.MaxValue or less than System.UInt16.MinValue.

1 **UInt16.Parse(System.String,** 2 **System.IFormatProvider) Method**

```
3    [ILAsm]  
4    .method public hidebysig static unsigned int16 Parse(string s, class  
5    System.IFormatProvider provider)  
  
6    [C#]  
7    public static ushort Parse(string s, IFormatProvider provider)
```

8 **Summary**

9 Returns the specified System.String converted to a System.UInt16 value.

10 **Type Attributes:**

- 11 · CLSCompliantAttribute(false)

12 **Parameters**

Parameter	Description
<i>s</i>	A System.String containing the value to convert. The string is interpreted using the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information about <i>s</i> .

15 **Return Value**

16 The System.UInt16 value obtained from *s*.

17 **Description**

18 This version of System.UInt16.Parse is equivalent to System.UInt16.Parse (*s*,
19 System.Globalization.NumberStyles.Integer, *provider*).

20 The string *s* is parsed using the culture-specific formatting information from the
21 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is
22 null or a System.Globalization.NumberFormatInfo cannot be obtained from *provider*,
23 the formatting information for the current system culture is used.
24
25
26

This method is not CLS-compliant. For a CLS-compliant alternative use `System.Int32.Parse(System.String, System.IFormatProvider)`.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct style.
System.OverflowException	s represents a number greater than <code>System.UInt16.MaxValue</code> or less than <code>System.UInt16.MinValue</code> .

UInt16.Parse(System.String, System.Globalization.NumberStyles, System.IFormatProvider) Method

```
[ILAsm]  
.method public hidebysig static unsigned int16 Parse(string s, valuetype  
System.Globalization.NumberStyles style, class System.IFormatProvider  
provider)  
  
[C#]  
public static ushort Parse(string s, NumberStyles style, IFormatProvider  
provider)
```

Summary

Returns the specified `System.String` converted to a `System.UInt16` value.

Type Attributes:

- `CLSCompliantAttribute(false)`

Parameters

Parameter	Description
<i>s</i>	A <code>System.String</code> containing the value to convert. The string is interpreted using the style specified by <i>style</i> .
<i>style</i>	Zero or more <code>System.Globalization.NumberStyles</code> values that specify the style of <i>s</i> . Specify multiple values for <i>style</i> using the bitwise OR operator. If <i>style</i> is a null reference, the string is interpreted using the <code>System.Globalization.NumberStyles.Integer</code> style.
<i>provider</i>	A <code>System.IFormatProvider</code> that supplies a <code>System.Globalization.NumberFormatInfo</code> containing culture-specific formatting information about <i>s</i> .

Return Value

The `System.UInt16` value obtained from *s*.

Description

The string *s* is parsed using the culture-specific formatting information from the `System.Globalization.NumberFormatInfo` instance supplied by *provider*. If *provider* is null or a `System.Globalization.NumberFormatInfo` cannot be obtained from *provider*, the formatting information for the current system culture is used.

This method is not CLS-compliant. For a CLS-compliant alternative use `System.Int32.Parse(System.String, System.Globalization.NumberStyles, System.IFormatProvider)`.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.FormatException	<i>s</i> is not in the correct style.
System.OverflowException	<i>s</i> represents a number greater than <code>System.UInt16.MaxValue</code> or less than <code>System.UInt16.MinValue</code> .

UInt16.ToString(System.IFormatProvider)

Method

```
[ILAsm]  
.method public final hidebysig virtual string ToString(class  
System.IFormatProvider provider)  
  
[C#]  
public string ToString(IFormatProvider provider)
```

Summary

Returns a `System.String` representation of the value of the current instance.

Parameters

Parameter	Description
<i>provider</i>	A <code>System.IFormatProvider</code> that supplies a <code>System.Globalization.NumberFormatInfo</code> containing culture-specific formatting information.

Return Value

A `System.String` representation of the current instance formatted using the general format specifier, ("G"). The string takes into account the formatting information in the `System.Globalization.NumberFormatInfo` instance supplied by *provider*.

Description

This version of `System.UInt16.ToString` is equivalent to `System.UInt16.ToString("G", provider)`.

If *provider* is null or a `System.Globalization.NumberFormatInfo` cannot be obtained from *provider*, the formatting information for the current system culture is used.

UInt16.ToString(System.String, System.IFormatProvider) Method

```
[ILAsm]  
.method public final hidebysig virtual string ToString(string format,  
class System.IFormatProvider provider)  
  
[C#]  
public string ToString(string format, IFormatProvider provider)
```

Summary

Returns a `System.String` representation of the value of the current instance.

Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> containing a character that specifies the format of the returned string.
<i>provider</i>	A <code>System.IFormatProvider</code> that supplies a <code>System.Globalization.NumberFormatInfo</code> instance containing culture-specific formatting information.

Return Value

A `System.String` representation of the current instance formatted as specified by *format*. The string takes into account the formatting information in the `System.Globalization.NumberFormatInfo` instance supplied by *provider*.

Description

If *provider* is null or a `System.Globalization.NumberFormatInfo` cannot be obtained from *provider*, the formatting information for the current system culture is used.

If *format* is a null reference, the general format specifier "G" is used.

[Note: For a detailed description of formatting, see the `System.IFormattable` interface.

This method is implemented to support the `System.IFormattable` interface.

]

The following table lists the characters that are valid for the `System.UInt16` type.

Format Characters	Description
"C", "c"	Currency format.
"D", "d"	Decimal format.
"E", "e"	Exponential notation format.
"F", "f"	Fixed-point format.
"G", "g"	General format.
"N", "n"	Number format.
"P", "p"	Percent format.
"X", "x"	Hexadecimal format.

Exceptions

Exception	Condition
System.FormatException	<i>format</i> is invalid.

1 UInt16.ToString() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual string ToString()  
  
4 [C#]  
5 public override string ToString()
```

6 Summary

7 Returns a `System.String` representation of the value of the current instance.

8 Return Value

9
10 A `System.String` representation of the current instance formatted using the general
11 format specifier ("G"). The string takes into account the current system culture.

12 Description

13 This version of `System.UInt16.ToString` is equivalent to
14 `System.UInt16.ToString(null, null)`.
15

16 [*Note:* This method overrides `System.Object.ToString`.]
17
18
19

UInt16.ToString(System.String) Method

```
[ILAsm]  
.method public hidebysig instance string ToString(string format)  
  
[C#]  
public string ToString(string format)
```

Summary

Returns a `System.String` representation of the value of the current instance.

Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies the format of the returned string. [Note: For a list of valid values, see <code>System.UInt16.ToString(System.String, System.IFormatProvider)</code> .]

Return Value

A `System.String` representation of the current instance formatted as specified by *format*. The string takes into account the current system culture.

Description

This method is equivalent to `System.UInt16.ToString(format, null)`.

If *format* is a null reference, the general format specifier "G" is used.

Exceptions

Exception	Condition
<code>System.FormatException</code>	<i>format</i> is invalid.

Example

```

1      This example demonstrates converting a System.UInt16 to a string.
2
3      [C#]

4      using System;
5      public class UInt16ToStringExample {
6          public static void Main() {
7              UInt16 i = 16;
8              Console.WriteLine(i);
9              String[] formats = {"c", "d", "e", "f", "g", "n", "p", "x" };
10             foreach(String str in formats)
11                 Console.WriteLine("{0}: {1}", str, i.ToString(str));
12         }
13     }

```

14 The output is

```

15
16 16
17
18
19 c: $16.00
20
21
22 d: 16
23
24
25 e: 1.600000e+001
26
27
28 f: 16.00
29
30
31 g: 16
32
33
34 n: 16.00
35
36
37 p: 1,600.00 %
38
39
40 x: 10
41

```

42