

# System.UInt32 Structure

```
[ILAsm]
.class public sequential sealed serializable UInt32 extends
System.ValueType implements System.IComparable, System.IFormattable,
System.IComparable`1<unsigned int32>, System.IEquatable`1<unsigned int32>

[C#]
public struct UInt32: IComparable, IFormattable, IComparable<UInt32>,
IEquatable<UInt32>
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 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.UInt32>**
- **System.IEquatable<System.UInt32>**

## Summary

Represents a 32-bit unsigned integer.

## Inherits From: System.ValueType

**Library:** BCL

**Thread Safety:** This type is safe for multithreaded operations.

## Description

The `System.UInt32` data type represents integer values ranging from 0 to positive 4,294,967,295 (hexadecimal 0xFFFFFFFF).

# UInt32.MaxValue Field

```
[ILAsm]  
.field public static literal unsigned int32 MaxValue = 4294967295  
  
[C#]  
public const uint MaxValue = 4294967295
```

## Summary

Contains the maximum value for the `System.UInt32` type.

## Description

The value of this constant is 4,294,967,295 (hexadecimal 0xFFFFFFFF).

# UInt32.MinValue Field

```
[ILAsm]  
.field public static literal unsigned int32 MinValue = 0  
  
[C#]  
public const uint MinValue = 0
```

## Summary

Contains the minimum value for the `System.UInt32` type.

## Description

The value of this constant is 0.

# UInt32.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.]

## Exceptions

Exception	Condition
-----------	-----------

**System.ArgumentException**

*value* is not a *System.UInt32* and is not a null reference.

1

2

# UInt32.CompareTo(System.UInt32) Method

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

## Summary

Returns the sort order of the current instance compared to the specified `System.UInt32`.

## Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt32</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<UInt32>` interface.]

# UInt32.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 type and value.

## 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 type and value as the current instance. If *obj* is a null reference or is not an instance of `System.UInt32`, returns `false`.

## Description

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

# UInt32.Equals(System.UInt32) Method

```
[ILAsm]  
.method public hidebysig virtual bool Equals(unsigned int32 obj)  
  
[C#]  
public override bool Equals(uint obj)
```

## Summary

Determines whether the current instance and the specified `System.UInt32` represent the same value.

## Parameters

Parameter	Description
<i>obj</i>	The <code>System.UInt32</code> to compare to the current instance.

## Return Value

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

## Description

[*Note:* This method is implemented to support the `System.IEquatable<UInt32>` interface.]



# UInt32.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 UInt32.Parse(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 Parse(string s)  
4 [C#]  
5 public static uint Parse(string s)
```

## 6 Summary

7 Returns the specified `System.String` converted to a `System.UInt32` value.

## 8 Type Attributes:

- 9 • `CLSCompliantAttribute(false)`

## 10 Parameters

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

## 12 Return Value

13 The `System.UInt32` value obtained from `s`.

## 14 Description

15 This version of `System.UInt32.Parse` is equivalent to `System.UInt32.Parse(s, System.Globalization.NumberStyles.Integer, null)`.

16 The string `s` is parsed using the formatting information in a `System.Globalization.NumberFormatInfo` initialized for the current system culture.  
17  
18 [Note: For more information, see `System.Globalization.NumberFormatInfo.CurrentInfo`.]  
19  
20  
21  
22  
23

24 This method is not CLS-compliant. For a CLS-compliant alternative use `System.Int64.Parse(System.String)`.  
25  
26

## 27 Exceptions

Exception	Condition
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct style.
<b>System.OverflowException</b>	s represents a number greater than System.UInt32.MaxValue or less than System.UInt32.MinValue.

1

## 2 Example

3 This example demonstrates parsing a string to a System.UInt32.

4

5 [C#]

```

6 using System;
7 public class UInt32ParseClass {
8     public static void Main() {
9         string str = " 100 ";
10        Console.WriteLine("String: \"{0}\" <UInt32> {1}",str,UInt32.Parse(str));
11    }
12 }
```

13 The output is

14

15 String: " 100 " <UInt32> 100

16

# UInt32.Parse(System.String, System.Globalization.NumberStyles) Method

```
[ILAsm]  
.method public hidebysig static unsigned int32 Parse(string s, valuetype  
System.Globalization.NumberStyles style)
```

```
[C#]  
public static uint Parse(string s, NumberStyles style)
```

## Summary

Returns the specified `System.String` converted to a `System.UInt32` 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.UInt32` value obtained from *s*.

## Description

This version of `System.UInt32.Parse` is equivalent to `System.UInt32.Parse(s, style, null)`.

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

1 This method is not CLS-compliant. For a CLS-compliant alternative use  
2 `System.Int64.Parse(System.String, System.Globalization.NumberStyles).`

3 **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct style.
<b>System.OverflowException</b>	s represents a number greater than <code>System.UInt32.MaxValue</code> or less than <code>System.UInt32.MinValue</code> .

4

5

# 1    **UInt32.Parse(System.String,** 2    **System.IFormatProvider) Method**

```
3    [ILAsm]  
4    .method public hidebysig static unsigned int32 Parse(string s, class  
5    System.IFormatProvider provider)  
  
6    [C#]  
7    public static uint Parse(string s, IFormatProvider provider)
```

## 8    **Summary**

9       Returns the specified `System.String` converted to a `System.UInt32` value.

## 10   **Type Attributes:**

- 11       • `CLSCompliantAttribute(false)`

## 12   **Parameters**

Parameter	Description
<i>s</i>	A <code>System.String</code> containing the value to convert. 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> .

## 14   **Return Value**

15       The `System.UInt32` value obtained from *s*.

## 16   **Description**

17       This version of `System.UInt32.Parse` is equivalent to `System.UInt32.Parse(s,`  
18       `System.Globalization.NumberStyles.Integer, provider)`.

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

23       This method is not CLS-compliant. For a CLS-compliant alternative use  
24       `System.Int64.Parse (System.String, System.IFormatProvider)`.

## 27   **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct style.
<b>System.OverflowException</b>	s represents a number greater than <code>System.UInt32.MaxValue</code> or less than <code>System.UInt32.MinValue</code> .

1

2

# UInt32.Parse(System.String, System.Globalization.NumberStyles, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 Parse(string s, valuetype
System.Globalization.NumberStyles style, class System.IFormatProvider
provider)

[C#]
public static uint Parse(string s, NumberStyles style, IFormatProvider
provider)
```

## Summary

Returns the specified System.String converted to a System.UInt32 value.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>s</i>	A System.String containing the value to convert. The string is interpreted using the style specified by <i>style</i> .
<i>style</i>	Zero or more System.Globalization.NumberStyles 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 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> .

## Return Value

The System.UInt32 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



1 null or a `System.Globalization.NumberFormatInfo` cannot be obtained from *provider*,  
2 the formatting information for the current system culture is used.  
3  
4 This method is not CLS-compliant. For a CLS-compliant alternative use  
5 `System.Int64.Parse(System.String, System.Globalization.NumberStyles,`  
6 `System.IFormatProvider)`.

7 **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct style.
<b>System.OverflowException</b>	s represents a number greater than <code>System.UInt32.MaxValue</code> or less than <code>System.UInt32.MinValue</code> .

8  
9

# UInt32.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.UInt32.ToString` is equivalent to `System.UInt32.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.

# UInt32.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.UInt32` 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.

1

## 2 Exceptions

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

3

4

# 1 UInt32.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 A `System.String` representation of the current instance formatted using the general  
10 format specifier, ("G"). The string takes into account the current system culture.

## 11 Description

12 This version of `System.UInt32.ToString` is equivalent to `System.UInt32.ToString`  
13 `(null, null)`.

14  
15 [*Note:* This method overrides `System.Object.ToString`.]  
16  
17

# UInt32.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. [ <i>Note:</i> For a list of valid values, see <code>System.UInt32.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.UInt32.ToString (format, null)`.

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

## Exceptions

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

## Example

This example demonstrates converting a `System.UInt32` to a string.

```
[C#]  
  
using System;  
public class UInt32ToStringExample {  
    public static void Main() {
```

```
1      UInt32 i = 32;
2      Console.WriteLine(i);
3      String[] formats = {"c", "d", "e", "f", "g", "n", "p", "x" };
4      foreach(String str in formats)
5          Console.WriteLine("{0}: {1}", str, i.ToString(str));
6      }
7  }
```

8 The output is

```
9
10 32
11
12
13 c: $32.00
14
15
16 d: 32
17
18
19 e: 3.200000e+001
20
21
22 f: 32.00
23
24
25 g: 32
26
27
28 n: 32.00
29
30
31 p: 3,200.00 %
32
33
34 x: 20
35
```

36