

System.Byte Structure

```
[ILASM]
.class public sequential sealed serializable Byte extends
System.ValueType implements System.IComparable,
System.IFormattable

[C#]
public struct Byte: IComparable, IFormattable
```

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)

Implements:

- **System.IComparable**
- **System.IFormattable**

Summary

Represents an 8-bit unsigned integer.

Inherits From: System.ValueType

Library: BCL

Thread Safety: This type is safe for multithreaded operations.

Description

The **System.Byte** data type represents integer values ranging from 0 to positive 255 (hexadecimal 0xFF).

1 Byte.MaxValue Field

```
2 [ILASM]  
3 .field public static literal unsigned int8 MaxValue = 255  
4 [C#]  
5 public const byte MaxValue = 255
```

6 Summary

7 Contains the maximum value for the **System.Byte** type.

8 Description

9 The value of this constant is 255 (hexadecimal 0XFF).

10

1 Byte.MinValue Field

```
2 [ILASM]  
3 .field public static literal unsigned int8 MinValue = 0  
4 [C#]  
5 public const byte MinValue = 0
```

6 Summary

7 Contains the minimum value for the **System.Byte** type.

8 Description

9 The value of this constant is 0.

10

1 Byte.CompareTo(System.Object) Method

```
2 [ILASM]  
3 .method public final hidebysig virtual int32  
4 CompareTo(object value)  
  
5 [C#]  
6 public int CompareTo(object value)
```

7 Summary

8 Returns the sort order of the current instance compared to the
9 specified object.

10 Parameters

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

14 Return Value

16 A **System.Int32** containing a value that reflects the sort order of the
17 current instance as compared to *value*. The following table defines the
18 conditions under which the return value is a negative number, zero, or
19 a positive number.

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

21 Description

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

24 Exceptions

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

1
2
3

1 Byte.Equals(System.Object) Method

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

6 Summary

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

9 Parameters

10
11

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

12
13
14

Return Value

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

18 Description

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

1 Byte.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** containing the 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 Byte.Parse(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int8 Parse(string  
4 s)  
5  
6 [C#]  
7 public static byte Parse(string s)
```

7 Summary

8 Returns the specified **System.String** converted to a **System.Byte**
9 value.

10 Parameters

11
12

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

13
14
15

14 Return Value

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

17 Description

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

21
22 The string s is parsed using the formatting information in a
23 **System.Globalization.NumberFormatInfo** initialized for the current
24 system culture. [Note: For more information, see
25 **System.Globalization.NumberFormatInfo.CurrentInfo**.]

26 Exceptions

27
28

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.Byte.MaxValue or less than System.Byte.MinValue .

1

2 **Example**

3

4 The following example demonstrates the **System.Byte.Parse** method.

5

6

```
[C#]
```

7

```
using System;
```

8

```
public class ByteParseClass {
```

9

```
public static void Main() {
```

10

```
    string str = " 100  ";
```

11

```
    Console.WriteLine("String: \"{0}\" <Byte>
```

12

```
{1}",str,Byte.Parse(str));
```

13

```
}
```

14

```
}
```

15

16

The output is

17

18

```
String: " 100 " <Byte> 100
```

19

1 Byte.Parse(System.String, 2 System.Globalization.NumberStyles) 3 Method

```
4 [ILASM]  
5 .method public hidebysig static unsigned int8 Parse(string  
6 s, valuetype System.Globalization.NumberStyles style)  
  
7 [C#]  
8 public static byte Parse(string s, NumberStyles style)
```

9 Summary

10 Returns the specified **System.String** converted to a **System.Byte**
11 value.

12 Parameters

13
14

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

15
16

16 Return Value

17

18 The **System.Byte** value obtained from *s*.

19 Description

20 This version of **System.Byte.Parse** is equivalent to
21 **System.Byte.Parse** (*s*, *style*, **null**).

22

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

27 Exceptions

28
29

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.

- 1
- 2
- 3

System.FormatException	s is not in the correct style.
System.OverflowException	s represents a number greater than System.Byte.MaxValue or less than System.Byte.MinValue .

1 Byte.Parse(System.String, 2 System.IFormatProvider) Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int8 Parse(string  
5 s, class System.IFormatProvider provider)  
  
6 [C#]  
7 public static byte Parse(string s, IFormatProvider  
8 provider)
```

9 Summary

10 Returns the specified **System.String** converted to a **System.Byte**
11 value.

12 Parameters

13
14

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
16

16 Return Value

17

18 The **System.Byte** value obtained from *s*.

19 Description

20 This version of **System.Byte.Parse** is equivalent to
21 **System.Byte.Parse** (*s*,
22 **System.Globalization.NumberStyles.Integer**, *provider*).

23

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

30 Exceptions

31
32

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

- 1
- 2
- 3

1 Byte.Parse(System.String, 2 System.Globalization.NumberStyles, 3 System.IFormatProvider) Method

```
4 [ILASM]  
5 .method public hidebysig static unsigned int8 Parse(string  
6 s, valuetype System.Globalization.NumberStyles style, class  
7 System.IFormatProvider provider)
```

```
8 [C#]  
9 public static byte Parse(string s, NumberStyles style,  
10 IFormatProvider provider)
```

11 Summary

12 Returns the specified **System.String** converted to a **System.Byte**
13 value.

14 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> .

17 Return Value

18 The **System.Byte** value obtained from *s*.

21 Description

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

1 **Exceptions**
2
3

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.Byte.MaxValue or less than System.Byte.MinValue .

4
5
6

1 Byte.ToString(System.IFormatProvider)

2 Method

```
3 [ILASM]  
4 .method public final hidebysig virtual string  
5 ToString(class System.IFormatProvider provider)  
  
6 [C#]  
7 public string ToString(IFormatProvider provider)
```

8 Summary

9 Returns a **System.String** representation of the value of the current
10 instance.

11 Parameters

12
13

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

14
15
16

Return Value

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

21 Description

22 This version of **System.Byte.ToString** is equivalent to
23 **System.Byte.ToString**("G", *provider*).

24
25
26
27

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

28

1 Byte.ToString(System.String, 2 System.IFormatProvider) Method

```
3 [ILASM]  
4 .method public final hidebysig virtual string  
5 ToString(string format, class System.IFormatProvider  
6 provider)  
  
7 [C#]  
8 public string ToString(string format, IFormatProvider  
9 provider)
```

10 Summary

11 Returns a **System.String** representation of the value of the current
12 instance.

13 Parameters

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

16 Return Value

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

23 Description

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

27
28 If *format* is a **null** reference, the general format specifier "G" is used.

29
30 The following table lists the characters that are valid for the
31 **System.Byte** 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
3
4
5
6

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

This method is implemented to support the **System.IFormattable** interface.]

7
8
9

Exceptions

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

10
11
12

1 Byte.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
8 instance.

9 Return Value

10

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

14 Description

15 This version of **System.Byte.ToString** is equivalent to
16 **System.Byte.ToString (null, null)**.

17

18 [*Note:* This method overrides **System.Object.ToString**.]

19

1 Byte.ToString(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig instance string ToString(string  
4 format)  
  
5 [C#]  
6 public string ToString(string format)
```

7 Summary

8 Returns a **System.String** representation of the value of the current
9 instance.

10 Parameters

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

14 Return Value

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

19 Description

20 This version of **System.Byte.ToString** is equivalent to
21 **System.Byte.ToString** (*format*, **null**).

22 If *format* is **null**, the general format specifier "G" is used.
23

24 Exceptions

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

28 Example

30 The following example demonstrates the **System.Byte.ToString**
31 method.

```
32 [C#]  
33
```

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

14 The output is

```
15      8
16
17
18
19      c: $8.00
20
21
22      d: 8
23
24
25      e: 8.000000e+000
26
27
28      f: 8.00
29
30
31      g: 8
32
```

1
2 n: 8.00
3
4
5 p: 800.00 %
6
7
8 x: 8
9
10