

# 1 System.SByte Structure

2  
3

```
4 [ILASM]  
5 .class public sequential sealed serializable SByte extends  
6 System.ValueType implements System.IComparable,  
7 System.IFormattable  
  
8 [C#]  
9 public struct SByte: IComparable, IFormattable
```

## 10 Assembly Info:

- 11 • Name: mscorlib
- 12 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 13 • Version: 1.0.x.x
- 14 • Attributes:
  - 15 ○ CLSCompliantAttribute(true)

## 16 Type Attributes:

- 17 • CLSCompliantAttribute(false)

## 18 Implements:

- 19 • System.IComparable
- 20 • System.IFormattable

## 21 Summary

22

23 Represents an 8-bit signed integer.

## 24 Inherits From: System.ValueType

25

26 Library: BCL

27

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

29

## 30 Description

31 The **System.SByte** data type represents integer values ranging from  
32 negative 128 to positive 127; that is, hexadecimal 0x80 to 0x7F.

33

# 1 SByte.MaxValue Field

```
2 [ILASM]  
3 .field public static literal int8 MaxValue = 127  
4 [C#]  
5 public const sbyte MaxValue = 127
```

## 6 Summary

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

## 8 Description

9 The value of this constant is 127 (hexadecimal 0X7F).

10

# 1 SByte.MinValue Field

```
2 [ILASM]  
3 .field public static literal int8 MinValue = -128  
4 [C#]  
5 public const sbyte MinValue = -128
```

## 6 Summary

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

## 8 Description

9 The value of this constant is -128 (hexadecimal 0X80).

10

# 1 SByte.CompareTo(System.Object) Method

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

## 7 Summary

8 Returns the sort order of the current instance compared to the  
9 specified **System.Object**.

## 10 Parameters

Parameter	Description
<i>obj</i>	The <b>System.Object</b> 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 *obj*. 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>obj</i> .
Zero	Current instance == <i>obj</i> .
Any positive number	Current instance > <i>obj</i> , or <i>obj</i> is a null reference.

## 21 Description

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

## 24 Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>obj</i> is not a <b>System.SByte</b> and is not a null reference.

1  
2  
3

# 1 SByte.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 <b>System.Object</b> 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.SByte**, returns **false**.

## 18 Description

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

# 1 SByte.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 SByte.Parse(System.String) Method

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

## 6 Summary

7 Returns the specified **System.String** converted to a **System.SByte**  
8 value.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

12  
13

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

14  
15  
16

## 15 Return Value

17 The **System.SByte** value obtained from s.

## 18 Description

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

22  
23  
24  
25  
26  
27

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 **System.Globalization.NumberFormatInfo.CurrentInfo**.]

28  
29

This method is not CLS-compliant. For a CLS-compliant alternative use **System.Int16.Parse** (**System.String**).

## 30 Exceptions

31  
32

Exception	Condition
<b>System.ArgumentException</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 <b>System.SByte.MaxValue</b> or less than <b>System.SByte.MinValue</b> .

1  
2  
3

## Example

4  
5  
6  
7

This example demonstrates the **System.SByte.Parse(System.String)** method.

[C#]

8  
9  
10  
11  
12  
13  
14  
15

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

16  
17  
18

The output is

String: " 100 " <SByte> 100

19

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

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

## 9 Summary

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

## 12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

## 14 Parameters

15  
16

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

17  
18  
19

## 18 Return Value

20 The **System.SByte** value obtained from *s*.

## 21 Description

22 This version of **System.SByte.Parse** is equivalent to  
23 **System.SByte.Parse** (*s*, *style*, **null**).

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

29  
30 This method is not CLS-compliant. For a CLS-compliant alternative use

1 **System.Int16.Parse(System.String,**  
2 **System.Globalization.NumberStyles).**

3 **Exceptions**

4  
5

<b>Exception</b>	<b>Condition</b>
<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 <b>System.SByte.MaxValue</b> or less than <b>System.SByte.MinValue</b> .

6  
7  
8

# SByte.Parse(System.String, System.IFormatProvider) Method

```
[ILASM]
.method public hidebysig static int8 Parse(string s, class
System.IFormatProvider provider)

[C#]
public static sbyte Parse(string s, IFormatProvider
provider)
```

## Summary

Returns the specified String converted to a **System.SByte** value.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

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

## Return Value

The **System.SByte** value obtained from *s*.

## Description

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

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.

1 This method is not CLS-compliant. For a CLS-compliant alternative use  
2 **System.Int16.Parse** (**System.String**, **System.IFormatProvider**).

3 **Exceptions**  
4  
5

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 <b>System.SByte.MaxValue</b> or less than <b>System.SByte.MinValue</b> .

6  
7  
8

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

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

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

## 11 Summary

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

## 14 Type Attributes:

- 15 • CLSCompliantAttribute(false)

## 16 Parameters

17  
18

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

19  
20  
21

## 20 Return Value

22 The **System.SByte** value obtained from *s*.

## 23 Description

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

1 *provider*, the formatting information for the current system culture is  
2 used.

3  
4 This method is not CLS-compliant. For a CLS-compliant alternative use  
5 **System.Int16.Parse(System.String,**  
6 **System.Globalization.NumberStyles, System.IFormatProvider).**

7 **Exceptions**

8  
9

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 <b>System.SByte.MaxValue</b> or less than <b>System.SByte.MinValue</b> .

10  
11  
12

# 1 SByte.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 <b>System.IFormatProvider</b> that supplies a <b>System.Globalization.NumberFormatInfo</b> 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 formatting information in the  
20 **System.Globalization.NumberFormatInfo** instance supplied by  
21 *provider*.

### 22 Description

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

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

29

# SByte.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 <b>System.String</b> containing a character that specifies the format of the returned string.
<i>provider</i>	A <b>System.IFormatProvider</b> that supplies a <b>System.Globalization.NumberFormatInfo</b> 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.SByte** 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**

3

4

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

5

6

7

# 1 SByte.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.SByte.ToString** is equivalent to  
16 **System.SByte.ToString (null, null)**.

17

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

19

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

## 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 version of **System.SByte.ToString** is equivalent to **System.SByte.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 the **System.SByte.ToString(System.String)** method.

```
[C#]
```

```
1      using System;
2      public class SByteToStringExample {
3          public static void Main() {
4              SByte i = 8;
5              Console.WriteLine(i);
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                     i.ToString(str));
11          }
12     }
```

13 The output is

14 8

15

16

17

18 c: \$8.00

19

20

21 d: 8

22

23

24 e: 8.000000e+000

25

26

27 f: 8.00

28

29

30 g: 8

31

32

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