

1 System.SByte Structure

```
2 [ILAsm]
3 .class public sequential sealed serializable SByte extends
4 System.ValueType implements System.IComparable, System.IFormattable,
5 System.IComparable`1<int8>, System.IEquatable`1<int8>
6
7 [C#]
8 public struct SByte: IComparable, IFormattable, IComparable<SByte>,
9 IEquatable<SByte>
```

9 Assembly Info:

- 10 • *Name:* mscorlib
- 11 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • *Version:* 2.0.x.x
- 13 • *Attributes:*
 - 14 ○ CLSCompliantAttribute(true)

15 Type Attributes:

- 16 • CLSCompliantAttribute(false)

17 Implements:

- 18 • **System.IComparable**
- 19 • **System.IFormattable**
- 20 • **System.IComparable<System.SByte>**
- 21 • **System.IEquatable<System.SByte>**

22 Summary

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 negative 128 to
32 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 CompareTo(object obj)  
4 [C#]  
5 public int CompareTo(object obj)
```

6 Summary

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

8 Parameters

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

9 Return Value

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

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

15 Description

17 [Note: This method is implemented to support the `System.IComparable` interface.]
18
19

20 Exceptions

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

System.ArgumentException

obj is not a `System.SByte` and is not a null reference.

1

2

SByte.CompareTo(System.SByte) Method

```
[ILAsm]  
.method public final hidebysig virtual int32 CompareTo(int8 value)  
  
[C#]  
public int CompareTo(sbyte value)
```

Summary

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

Parameters

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

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 `System.Object` represent the
8 same type and value.

9 Parameters

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

10

11 Return Value

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

14 Description

15 [Note: This method overrides `System.Object.Equals`.]
16
17

18

1 SByte.Equals(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual bool Equals(int8 obj)  
4 [C#]  
5 public override bool Equals(sbyte obj)
```

6 Summary

7 Determines whether the current instance and the specified `System.SByte` represent the
8 same value.

9 Parameters

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

10

11 Return Value

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

13 Description

14 [Note: This method is implemented to support the `System.IEquatable<SByte>`
15 interface.]

16

17

18

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 A `System.Int32` containing the hash code for the current instance.

10 Description

11 The algorithm used to generate the hash code is unspecified.

12

13 [*Note:* This method overrides `System.Object.GetHashCode()`.]

14

15

16

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 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.

11

12 Return Value

13 The System.SByte value obtained from s.

14 Description

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

17

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

20 [Note: for more information, see
21 System.Globalization.NumberFormatInfo.CurrentInfo.]

22

23

24

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

27 Exceptions

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

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

1

2 Example

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

4

5 [C#]

```

6 using System;
7 public class SByteParseClass {
8     public static void Main() {
9         string str = " 100 ";
10        Console.WriteLine("String: \"{0}\" <SByte> {1}",
11                          str, SByte.Parse(str));
12    }
13 }

```

14 The output is

15

16 String: " 100 " <SByte> 100

17

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

```
[ILAsm]
.method public hidebysig static int8 Parse(string s, valuetype
System.Globalization.NumberStyles style)

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

Summary

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

Type Attributes:

- `CLSCompliantAttribute(false)`

Parameters

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

Return Value

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

Description

This version of `System.SByte.Parse` is equivalent to `System.SByte.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 `System.Globalization.NumberFormatInfo.CurrentInfo`.]

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

3 **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.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> .

4

5

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

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

8 Summary

9 Returns the specified String converted to a System.SByte 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> .

13

14 Return Value

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

16 Description

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

19

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

27 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.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> .

1

2

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

```
[ILAsm]
.method public hidebysig static int8 Parse(string s, valuetype
System.Globalization.NumberStyles style, class System.IFormatProvider
provider)

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

Summary

Returns the specified System.String converted to a System.SByte 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.SByte 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.Int16.Parse(System.String, System.Globalization.NumberStyles,`
6 `System.IFormatProvider)`.

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

8

9

1 SByte.ToString(System.IFormatProvider)

2 Method

```
3 [ILAsm]  
4 .method public final hidebysig virtual string ToString(class  
5 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 instance.

10 Parameters

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

11

12 Return Value

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

16 Description

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

19

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

22

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 <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.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

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

3

4

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 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.SByte.ToString` is equivalent to `System.SByte.ToString (null,`
13 `null)`.

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

18

1 SByte.ToString(System.String) Method

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

6 Summary

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

8 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.SByte.ToString (System.String, System.IFormatProvider)</code> .]

9

10 Return Value

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

13 Description

14 This version of `System.SByte.ToString` is equivalent to `System.SByte.ToString`
15 (*format*, `null`).

16

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

18 Exceptions

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

19

20 Example

21 This example demonstrates the `System.SByte.ToString(System.String)` method.

22

23 [C#]

```
24 using System;  
25 public class SByteToStringExample {
```

```
1     public static void Main() {
2         SByte i = 8;
3         Console.WriteLine(i);
4         String[] formats = {"c", "d", "e", "f", "g", "n", "p", "x"};
5         foreach(String str in formats)
6             Console.WriteLine("{0}: {1}", str, i.ToString(str));
7     }
8 }
```

9 The output is

```
10
11 8
12
13
14 c: $8.00
15
16
17 d: 8
18
19
20 e: 8.000000e+000
21
22
23 f: 8.00
24
25
26 g: 8
27
28
29 n: 8.00
30
31
32 p: 800.00 %
33
34
35 x: 8
36
```

37