

System.SByte Structure

```
[ILAsm]
.class public sequential sealed serializable SByte extends
System.ValueType implements System.IComparable, System.IFormattable,
System.IComparable`1<int8>, System.IEquatable`1<int8>

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

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.SByte>**
- **System.IEquatable<System.SByte>**

Summary

Represents an 8-bit signed integer.

Inherits From: System.ValueType

Library: BCL

Thread Safety: This type is safe for multithreaded operations.

Description

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

SByte.MaxValue Field

```
[ILAsm]  
.field public static literal int8 MaxValue = 127
```

```
[C#]  
public const sbyte MaxValue = 127
```

Summary

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

Description

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

SByte.MinValue Field

```
[ILAsm]  
.field public static literal int8 MinValue = -128  
  
[C#]  
public const sbyte MinValue = -128
```

Summary

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

Description

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

SByte.CompareTo(System.Object) Method

```
[ILAsm]  
.method public final hidebysig virtual int32 CompareTo(object obj)
```

```
[C#]  
public int CompareTo(object obj)
```

Summary

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

Parameters

| Parameter | Description |
|------------|--|
| <i>obj</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 *obj*. 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>obj</i> . |
| Zero | Current instance == <i>obj</i> . |
| A positive number | Current instance > <i>obj</i> , or <i>obj</i> is a null reference. |

Description

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

Exceptions

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

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.]

SByte.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.SByte`, returns `false`.

Description

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

SByte.Equals(System.SByte) Method

```
[ILAsm]  
.method public hidebysig virtual bool Equals(int8 obj)
```

```
[C#]  
public override bool Equals(sbyte obj)
```

Summary

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

Parameters

| Parameter | Description |
|------------|---|
| <i>obj</i> | The <code>System.SByte</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<SByte>` interface.]

SByte.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.`]

SByte.Parse(System.String) Method

```
[ILAsm]  
.method public hidebysig static int8 Parse(string s)
```

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

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 <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, System.Globalization.NumberStyles.Integer, 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`.]

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

Exceptions

| Exception | Condition |
|---------------------------------------|-------------------------------------|
| <code>System.ArgumentException</code> | <code>s</code> 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. |

Example

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

[C#]

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

The output is

```
String: " 100 " <SByte> 100
```

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 |
|--------------|--|
| <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.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`.]

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

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> . |

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 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> . |

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.

This method is not CLS-compliant. For a CLS-compliant alternative use System.Int16.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.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> . |

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 <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.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 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.Int16.Parse(System.String, System.Globalization.NumberStyles, 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.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> . |

SByte.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.SByte.ToString` is equivalent to `System.SByte.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.

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. |

Exceptions

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

SByte.ToString() Method

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

Summary

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

Return Value

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

Description

This version of `System.SByte.ToString` is equivalent to `System.SByte.ToString(null, null)`.

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

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 <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> .] |

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 |
|-------------------------------------|---------------------------|
| <code>System.FormatException</code> | <i>format</i> is invalid. |

Example

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

```
[C#]
```

```
using System;
```

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

The output is

8

c: \$8.00

d: 8

e: 8.000000e+000

f: 8.00

g: 8

n: 8.00

p: 800.00 %

x: 8