# System.SByte Structure

[ILASM]

```
.class public sequential sealed serializable SByte extends
System.ValueType implements System.IComparable,
System.IFormattable
```

[C#]

```
public struct SByte: 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)

**Type Attributes:**

- CLSCompliantAttribute(false)

**Implements:**

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

## 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 |
|---|---|
| *obj* | The **System.Object** to compare to the current instance. |

## Return Value

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

| Return Value | Description |
|---|---|
| Any negative number | Current instance < *obj*. |
| Zero | Current instance == *obj*. |
| Any positive number | Current instance > *obj*, or *obj* is a null reference. |

## Description

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentException** | *obj* is not a **System.SByte** and is not a null reference. |

# 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 |
|---|---|
| *obj* | The **System.Object** 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.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 |
|---|---|
| *s* | A **System.String** containing the value to convert. The string is interpreted using the **System.Globalization.NumberStyles.Integer** 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 |
|---|---|
| **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**. |

## 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 |
|---|---|
| *s* | A **System.String** containing the value to convert. The string is interpreted using the style specified by *style*. |
| *style* | Zero or more **System.Globalization.NumberStyles** values that specify the style of *s*. Specify multiple values for *style* using the bitwise OR operator. If *style* is a null reference, the string is interpreted using the **System.Globalization.NumberStyles.Integer** 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 **System.SByte.MaxValue** or less than **System.SByte.MinValue**. |

# 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 |
|---|---|
| *s* | A **System.String** containing the value to convert. The string is interpreted using the **System.Globalization.NumberStyles.Integer** style. |
| *provider* | A **System.IFormatProvider** that supplies a **System.Globalization.NumberFormatInfo** containing culture-specific formatting information about *s*. |

## 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 **System.SByte.MaxValue** or less than **System.SByte.MinValue**. |

# 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 |
|---|---|
| *s* | A **System.String** containing the value to convert. The string is interpreted using the style specified by *style*. |
| *style* | Zero or more **System.Globalization.NumberStyles** values that specify the style of *s*. Specify multiple values for *style* using the bitwise OR operator. If *style* is a null reference, the string is interpreted using the **System.Globalization.NumberStyles.Integer** style. |
| *provider* | A **System.IFormatProvider** that supplies a **System.Globalization.NumberFormatInfo** containing culture-specific formatting information about *s*. |

## 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 **System.SByte.MaxValue** or less than **System.SByte.MinValue**. |

# 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 |
|---|---|
| *provider* | A **System.IFormatProvider** that supplies a **System.Globalization.NumberFormatInfo** 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 |
|---|---|
| *format* | A **System.String** containing a character that specifies the format of the returned string. |
| *provider* | A **System.IFormatProvider** that supplies a **System.Globalization.NumberFormatInfo** 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 |
|---|---|
| **System.FormatException** | *format* 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 |
|---|---|
| *format* | A **System.String** that specifies the format of the returned string. [*Note:* For a list of valid values, see **System.SByte.ToString** (**System.String**, **System.IFormatProvider**).] |

## 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 |
|---|---|
| **System.FormatException** | *format* 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
```