

# System.Int16 Structure

```
[ILASM]
.class public sequential sealed serializable Int16 extends
System.ValueType implements System.IComparable,
System.IFormattable

[C#]
public struct Int16: IComparable, IFormattable
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Implements:

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

## Summary

Represents a 16-bit signed integer.

## Inherits From: System.ValueType

**Library:** BCL

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

## Description

The **System.Int16** data type represents integer values ranging from negative 32,768 to positive 32,767; that is, hexadecimal: 0x8000 to 0x7FFF.

# 1 Int16.MaxValue Field

```
2 [ILASM]  
3 .field public static literal int16 MaxValue = 32767  
4 [C#]  
5 public const short MaxValue = 32767
```

## 6 Summary

7 Contains the maximum value for the **System.Int16** type.

## 8 Description

9 The value of this constant is 32,767 (hexadecimal 0X7FFF).

10

# 1 Int16.MinValue Field

```
2 [ILASM]  
3 .field public static literal int16 MinValue = -32768  
4 [C#]  
5 public const short MinValue = -32768
```

## 6 Summary

7 Contains the minimum value for the **System.Int16** type.

## 8 Description

9 The value of this constant is -32,768 (hexadecimal 0X8000).

10

# Int16.CompareTo(System.Object) Method

```
[ILASM]
.method public final hidebysig virtual int32
CompareTo(object value)

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

## Summary

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

## Parameters

Parameter	Description
<i>value</i>	The <b>System.Object</b> 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 *value*. 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 < <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.

## Description

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

## Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is not a <b>System.Int16</b> and is not a null reference.

1  
2  
3

## Int16.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 <b>System.Object</b> 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.Int16**, returns **false**.

### Description

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

# 1 Int16.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4  
5 [C#]  
6 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

# Int16.Parse(System.String) Method

```
[ILASM]  
.method public hidebysig static int16 Parse(string s)  
  
[C#]  
public static short Parse(string s)
```

## Summary

Returns the specified **System.String** converted to a **System.Int16** value.

## Parameters

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.

## Return Value

The **System.Int16** value obtained from s.

## Description

This version of **System.Int16.Parse** is equivalent to **System.Int16.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**.]

## Exceptions

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.Int16.MaxValue</b> or less than <b>System.Int16.MinValue</b> .

1

## 2 **Example**

3

4 This example demonstrates parsing a string to a **System.Int16**.

5

6 [C#]

7 using System;

8 public class Int16ParseClass {

9 public static void Main() {

10 string str = " 100 ";

11 Console.WriteLine("String: \"{0}\" <Int16>

12 {1}",str,Int16.Parse(str));

13 }

14 }

15 The output is

16

17 String: " 100 " <Int16> 100

18

# Int16.Parse(System.String, System.Globalization.NumberStyles) Method

```
[ILASM]
.method public hidebysig static int16 Parse(string s,
valuetype System.Globalization.NumberStyles style)

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

## Summary

Returns the specified **System.String** converted to a **System.Int16** value.

## Parameters

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.

## Return Value

The **System.Int16** value obtained from *s*.

## Description

This version of **System.Int16.Parse** is equivalent to **System.Int16.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**.]

## Exceptions

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

1  
2  
3

<b>System.FormatException</b>	s is not in the correct style.
<b>System.OverflowException</b>	s represents a number greater than <b>System.Int16.MaxValue</b> or less than <b>System.Int16.MinValue</b> .

# Int16.Parse(System.String, System.IFormatProvider) Method

```
[ILASM]
.method public hidebysig static int16 Parse(string s, class
System.IFormatProvider provider)

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

## Summary

Returns the specified **System.String** converted to a **System.Int16** value.

## 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.Int16** value obtained from *s*.

## Description

This version of **System.Int16.Parse** is equivalent to **System.Int16.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.

## Exceptions

1  
2  
3

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.Int16.MaxValue</b> or less than <b>System.Int16.MinValue</b> .

# Int16.Parse(System.String, System.Globalization.NumberStyles, System.IFormatProvider) Method

```
[ILASM]  
.method public hidebysig static int16 Parse(string s,  
valuetype System.Globalization.NumberStyles style, class  
System.IFormatProvider provider)
```

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

## Summary

Returns the specified **System.String** converted to a **System.Int16** value.

## Parameters

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> .

## Return Value

The **System.Int16** 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.

1   **Exceptions**  
2  
3

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.Int16.MaxValue</b> or less than <b>System.Int16.MinValue</b> .

4  
5  
6

# Int16.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 <b>System.IFormatProvider</b> that supplies a <b>System.Globalization.NumberFormatInfo</b> 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.Int16.ToString** is equivalent to **System.Int16.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.

## Int16.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.Int16** 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 Int16.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.Int16.ToString** is equivalent to  
16 **System.Int16.ToString(null, null)**.

17

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

19

# Int16.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. [Note: For a list of valid values, see <b>System.Int16.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.Int16.ToString** is equivalent to **System.Int16.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 converting a **System.Int16** to a string.

```
[C#]
```

```

1      using System;
2      public class Int16ToStringExample {
3          public static void Main() {
4              Int16 i = 16;
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  
15 16

16  
17  
18 c: \$16.00

19  
20  
21 d: 16

22  
23  
24 e: 1.600000e+001

25  
26  
27 f: 16.00

28  
29  
30 g: 16

31  
32

1	n: 16.00
2	
3	
4	p: 1,600.00 %
5	
6	
7	x: 10
8	
9	