

# 1 System.Int32 Structure

2  
3

```
4 [ILASM]  
5 .class public sequential sealed serializable Int32 extends  
6 System.ValueType implements System.IComparable,  
7 System.IFormattable  
  
8 [C#]  
9 public struct Int32: 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 00]
- 13 • *Version:* 1.0.x.x
- 14 • *Attributes:*
  - 15 ○ CLSCompliantAttribute(true)

## 16 Implements:

- 17 • **System.IComparable**
- 18 • **System.IFormattable**

## 19 Summary

20

21 Represents a 32-bit signed integer.

## 22 Inherits From: System.ValueType

23

24 **Library:** BCL

25

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

27

## 28 Description

29 The **System.Int32** data type represents integer values ranging from  
30 negative 2,147,483,648 to positive 2,147,483,647; that is,  
31 hexadecimal 0X80000000 to 0X7FFFFFFF.

32

# 1 Int32.MaxValue Field

```
2 [ILASM]  
3 .field public static literal int32 MaxValue = 2147483647  
4 [C#]  
5 public const int MaxValue = 2147483647
```

## 6 Summary

7 Contains the maximum value for the **System.Int32** type.

## 8 Description

9 The value of this constant is 2,147,483,647 (hexadecimal  
10 0X7FFFFFFF).

11

# 1 Int32.MinValue Field

```
2 [ILASM]  
3 .field public static literal int32 MinValue = -2147483648  
4 [C#]  
5 public const int MinValue = -2147483648
```

## 6 Summary

7 Contains the minimum value for the **System.Int32** type.

## 8 Description

9 The value of this constant is 2,147,483,647 (hexadecimal  
10 0X7FFFFFFF).

11

# 1 Int32.CompareTo(System.Object) Method

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

## 7 Summary

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

## 10 Parameters

Parameter	Description
<i>value</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 *value*. 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>value</i> .
Zero	Current instance == <i>value</i> .
Any positive number	Current instance > <i>value</i> , or <i>value</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>value</i> is not a <b>System.Int32</b> and is not a null reference.

1  
2  
3

# 1 Int32.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.Int32**, returns **false**.

## 18 Description

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

# 1 Int32.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 Int32.Parse(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static int32 Parse(string s)  
4 [C#]  
5 public static int Parse(string s)
```

## 6 Summary

7 Returns the specified **System.String** converted to a **System.Int32**  
8 value.

## 9 Parameters

10  
11

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.

12  
13  
14

## Return Value

15 The **System.Int32** value obtained from s.

## 16 Description

17 This version of **System.Int32.Parse** is equivalent to  
18 **System.Int32.Parse(s,**  
19 **System.Globalization.NumberStyles.Integer, null)**.

20  
21  
22  
23  
24

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**.]

## 25 Exceptions

26  
27

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.Int32.MaxValue</b> or less than <b>System.Int32.MinValue</b> .

1

2 **Example**

3

4 This example demonstrates parsing a string to a **System.Int32**.

5

6

```
[C#]
```

7

```
using System;
```

8

```
public class Int32ParseClass {
```

9

```
    public static void Main() {
```

10

```
        string str = " 100  ";
```

11

```
        Console.WriteLine("String: \"{0}\" <Int32>
```

12

```
{1}",str,Int32.Parse(str));
```

13

```
    }
```

14

```
}
```

15

The output is

16

17

```
String: " 100 " <Int32> 100
```

18

# 1 Int32.Parse(System.String, 2 System.Globalization.NumberStyles) 3 Method

```
4 [ILASM]  
5 .method public hidebysig static int32 Parse(string s,  
6 valuetype System.Globalization.NumberStyles style)  
  
7 [C#]  
8 public static int Parse(string s, NumberStyles style)
```

## 9 Summary

10 Returns the specified **System.String** converted to a **System.Int32**  
11 value.

## 12 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.

## 15 Return Value

16 The **System.Int32** value obtained from *s*.

## 19 Description

20 This version of **System.Int32.Parse** is equivalent to  
21 **System.Int32.Parse** (*s*, *style*, **null**).

22  
23 The string *s* is parsed using the formatting information in a  
24 **System.Globalization.NumberFormatInfo** initialized for the current  
25 system culture. [*Note:* For more information, see  
26 **System.Globalization.NumberFormatInfo.CurrentInfo**.]

## 27 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.Int32.MaxValue</b> or less than <b>System.Int32.MinValue</b> .

# 1 Int32.Parse(System.String, 2 System.IFormatProvider) Method

```
3 [ILASM]  
4 .method public hidebysig static int32 Parse(string s, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static int Parse(string s, IFormatProvider provider)
```

## 8 Summary

9 Returns the specified **System.String** converted to a **System.Int32**  
10 value.

## 11 Parameters

12  
13

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> .

14  
15  
16

## Return Value

17 The **System.Int32** value obtained from *s*.

## 18 Description

19 This version of **System.Int32.Parse** is equivalent to  
20 **System.Int32.Parse**(*s*,  
21 **System.Globalization.NumberStyles.Integer**, *provider*).

22  
23 The string *s* is parsed using the culture-specific formatting information  
24 from the **System.Globalization.NumberFormatInfo** instance  
25 supplied by *provider*. If *provider* is **null** or a  
26 **System.Globalization.NumberFormatInfo** cannot be obtained from  
27 *provider*, the formatting information for the current system culture is  
28 used.

## 29 Exceptions

30  
31

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.Int32.MaxValue</b> or less than <b>System.Int32.MinValue</b> .

1  
2  
3

# 1 Int32.Parse(System.String, 2 System.Globalization.NumberStyles, 3 System.IFormatProvider) Method

```
4 [ILASM]  
5 .method public hidebysig static int32 Parse(string s,  
6 valuetype System.Globalization.NumberStyles style, class  
7 System.IFormatProvider provider)
```

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

## 11 Summary

12 Returns the specified **System.String** converted to a **System.Int32**  
13 value.

## 14 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> .

## 18 Return Value

20 The **System.Int32** value obtained from *s*.

## 21 Description

22 The string *s* is parsed using the culture-specific formatting information  
23 from the **System.Globalization.NumberFormatInfo** instance  
24 supplied by *provider*. If *provider* is **null** or a  
25 **System.Globalization.NumberFormatInfo** cannot be obtained from  
26 *provider*, the formatting information for the current system culture is  
27 used.

1 **Exceptions**  
2  
3

<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.Int32.MaxValue</b> or less than <b>System.Int32.MinValue</b> .

4  
5  
6

# 1 Int32.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.Int32.ToString** is equivalent to  
24 **System.Int32.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

# 1 Int32.ToString(System.String, 2 System.IFormatProvider) Method

```
3 [ILASM]  
4 .method public final hidebysig virtual string  
5 ToString(string format, class System.IFormatProvider  
6 provider)  
  
7 [C#]  
8 public string ToString(string format, IFormatProvider  
9 provider)
```

## 10 Summary

11 Returns a **System.String** representation of the value of the current  
12 instance.

## 13 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.

## 16 Return Value

17 A **System.String** representation of the current instance formatted as  
18 specified by *format*. The string takes into account the formatting  
19 information in the **System.Globalization.NumberFormatInfo**  
20 instance supplied by *provider*.  
21  
22

## 23 Description

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

28 If *format* is a null reference, the general format specifier "G" is used.  
29

30 [Note: For a detailed description of formatting, see the  
31 **System.IFormattable** interface.  
32

33 This method is implemented to support the **System.IFormattable**  
34 interface.] The following table lists the characters that are valid for the  
35 **System.Int32** type.

Item	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 Int32.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.Int32.ToString** is equivalent to  
16 **System.Int32.ToString (null, null)**.

17

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

19

# 1 Int32.ToString(System.String) Method

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

## 7 Summary

8 Returns a **System.String** representation of the value of the current  
9 instance.

## 10 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.Int32.ToString(System.String, System.IFormatProvider)</b> .]

## 14 Return Value

16 A **System.String** representation of the current instance formatted as  
17 specified by *format*. The string takes into account the current system  
18 culture.

## 19 Description

20 This method is equivalent to **System.Int32.ToString(*format*, null)**.

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

## 23 Exceptions

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

## 27 Example

29 This example demonstrates converting a **System.Int32** to a string.

```
30 [C#]  
31  
32 using System;
```

```
1 public class Int32ToStringExample {
2     public static void Main() {
3         Int32 i = 32;
4         Console.WriteLine(i);
5         String[] formats = {"c", "d", "e", "f", "g", "n",
6 "p", "x" };
7         foreach(String str in formats)
8             Console.WriteLine("{0}: {1}", str,
9 i.ToString(str));
10    }
11 }
```

12 The output is

```
13
14 32
15
16
17 c: $32.00
18
19
20 d: 32
21
22
23 e: 3.200000e+001
24
25
26 f: 32.00
27
28
29 g: 32
30
31
```

1 n: 32.00  
2  
3  
4 p: 3,200.00 %  
5  
6  
7 x: 20  
8  
9