

System.Int64 Structure

```
[ILAsm]
.class public sequential sealed serializable Int64 extends
System.ValueType implements System.IComparable, System.IFormattable,
System.IComparable`1<int64>, System.IEquatable`1<int64>

[C#]
public struct Int64: IComparable, IFormattable, IComparable<Int64>,
IEquatable<Int64>
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

- **System.IComparable**
- **System.IFormattable**
- **System.IComparable<System.Int64>**
- **System.IEquatable<System.Int64>**

Summary

Represents a 64-bit signed integer.

Inherits From: System.ValueType

Library: BCL

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

The `System.Int64` data type represents integer values ranging from negative 9,223,372,036,854,775,808 to positive 9,223,372,036,854,775,807; that is, hexadecimal 0X8000000000000000 to 0X7FFFFFFFFFFFFFFF.

Int64.MaxValue Field

```
[ILAsm]  
.field public static literal int64 MaxValue = 9223372036854775807  
  
[C#]  
public const long MaxValue = 9223372036854775807
```

Summary

Contains the maximum value for the `System.Int64` type.

Description

The value of this constant is 9,223,372,036,854,775,807 (hexadecimal 0X7FFFFFFFFFFFFFFF).

Int64.MinValue Field

```
[ILAsm]  
.field public static literal int64 MinValue = -9223372036854775808  
  
[C#]  
public const long MinValue = -9223372036854775808
```

Summary

Contains the minimum value for the `System.Int64` type.

Description

The value of this constant is -9,223,372,036,854,775,808 (hexadecimal 0X8000000000000000).

Int64.CompareTo(System.Int64) Method

```
[ILAsm]  
.method public final hidebysig virtual int32 CompareTo(int64 value)  
  
[C#]  
public int CompareTo(long value)
```

Summary

Returns the sort order of the current instance compared to the specified `System.Int64`.

Parameters

Parameter	Description
<i>value</i>	The <code>System.Int64</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<Int64>` interface.]

Int64.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 <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 *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> , or <i>value</i> is a null reference.

Description

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

1 **Exceptions**

2

3

Exception	Condition
System.ArgumentException	<i>value</i> is not a <code>System.Int64</code> and is not a null reference.

4

5

6

Int64.Equals(System.Int64) Method

```
[ILAsm]  
.method public hidebysig virtual bool Equals(int64 obj)  
  
[C#]  
public override bool Equals(long obj)
```

Summary

Determines whether the current instance and the specified `System.Int64` represent the same value.

Parameters

Parameter	Description
<i>obj</i>	The <code>System.Int64</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<Int64>` interface.]

Int64.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.Int64`, returns `false`.

Description

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

Int64.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()`.]

Int64.Parse(System.String) Method

```
[ILAsm]  
.method public hidebysig static int64 Parse(string s)  
  
[C#]  
public static long Parse(string s)
```

Summary

Returns the specified `System.String` converted to a `System.Int64` value.

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.Int64` value obtained from `s`.

Description

This version of `System.Int64.Parse` is equivalent to `System.Int64.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
<code>System.ArgumentNullException</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.Int64.MaxValue or less than System.Int64.MinValue.

Example

This example demonstrates parsing a string to a System.Int64.

[C#]

```
using System;
public class Int64ParseClass {
    public static void Main() {
        string str = " 100 ";
        Console.WriteLine("String: \"{0}\" <Int64> {1}", str, Int64.Parse(str));
    }
}
```

The output is

```
String: " 100 " <Int64> 100
```

Int64.Parse(System.String, System.Globalization.NumberStyles) Method

```
[ILAsm]  
.method public hidebysig static int64 Parse(string s, valuetype  
System.Globalization.NumberStyles style)  
  
[C#]  
public static long Parse(string s, NumberStyles style)
```

Summary

Returns the specified `System.String` converted to a `System.Int64` value.

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.Int64` value obtained from *s*.

Description

This version of `System.Int64.Parse` is equivalent to `System.Int64.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
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.Int64.MaxValue</code> or less than <code>System.Int64.MinValue</code> .

Int64.Parse(System.String, System.IFormatProvider) Method

```
[ILAsm]  
.method public hidebysig static int64 Parse(string s, class  
System.IFormatProvider provider)  
  
[C#]  
public static long Parse(string s, IFormatProvider provider)
```

Summary

Returns the specified `System.String` converted to a `System.Int64` value.

Parameters

Parameter	Description
<i>s</i>	A <code>System.String</code> containing the value to convert. 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.Int64` value obtained from *s*.

Description

This version of `System.Int64.Parse` is equivalent to `System.Int64.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

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.Int64.MaxValue</code> or less than <code>System.Int64.MinValue</code> .

1
2
3

Int64.Parse(System.String, System.Globalization.NumberStyles, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static int64 Parse(string s, valuetype
System.Globalization.NumberStyles style, class System.IFormatProvider
provider)

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

Summary

Returns the specified System.String converted to a System.Int64 value.

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.Int64 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 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.Int64.MaxValue</code> or less than <code>System.Int64.MinValue</code> .

Int64.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.Int64.ToString` is equivalent to `System.Int64.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.

Int64.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.Int64` 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	<i>format</i> is invalid.

1 Int64.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
10 A `System.String` representation of the current instance formatted using the general
11 format specifier ("G"). The string takes into account the current system culture.

12 Description

13 This version of `System.Int64.ToString` is equivalent to `System.Int64.ToString(null,`
14 `null)`.

15
16 [*Note:* This method overrides `System.Object.ToString`.]
17
18
19

Int64.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. [Note: For a list of valid values, see <code>System.Int64.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 method is equivalent to `System.Int64.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

```
1      This example demonstrates converting a System.Int64 to a string.
2
3      [C#]
4
5      using System;
6      public class Int64ToStringExample {
7          public static void Main() {
8              Int64 i = 64;
9              Console.WriteLine(i);
10             String[] formats = {"c", "d", "e", "f", "g", "n", "p", "x" };
11             foreach(String str in formats)
12                 Console.WriteLine("{0}: {1}", str, i.ToString(str));
13         }
14     }
```

14 The output is

```
15
16 64
17
18
19 c: $64.00
20
21
22 d: 64
23
24
25 e: 6.400000e+001
26
27
28 f: 64.00
29
30
31 g: 64
32
33
34 n: 64.00
35
36
37 p: 6,400.00 %
38
39
40 x: 40
41
```

42