

# 1 System.Convert Class

```
2 [ILAsm]  
3 .class public sealed Convert extends System.Object  
  
4 [C#]  
5 public sealed class Convert
```

## 6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
  - 11 ○ CLSCompliantAttribute(true)

## 12 Summary

13 Performs conversions between base data types.

## 14 Inherits From: System.Object

15

16 **Library:** BCL

17

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

20

## 21 Description

22 The following table shows conversions from source types to destination types. The first  
23 column contains the source types. The remaining columns indicate the destination types  
24 the source can be converted to. An 'x' indicates the *System.Convert* class implements  
25 the conversion. [*Note:* the column headers correspond precisely, in order, to the source  
26 types in the first column, but have been abbreviated to fit.]

27

28

Type	Bool	Byte	Char	DT	Dec	Dou	I16	I32	I64	SBy	Sin	Str	UI16	UI32	UI64
Boolean	x	x			x	x	x	x	x	x	x	x	x	x	x
Byte	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Char		x	x				x	x	x	x		x	x	x	x
DateTime				x								x			

Decimal	x	x			x	x	x	x	x	x	x	x	x	x	x
Double	x	x			x	x	x	x	x	x	x	x	x	x	x
Int16	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Int32	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Int64	x	x	x		x	x	x	x	x	x	x	x	x	x	x
SByte	x	x	x		x	x	x	x	x	x	x	x	x	x	x
Single	x	x			x	x	x	x	x	x	x	x	x	x	x
String	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
UInt16	x	x	x		x	x	x	x	x	x	x	x	x	x	x
UInt32	x	x	x		x	x	x	x	x	x	x	x	x	x	x
UInt64	x	x	x		x	x	x	x	x	x	x	x	x	x	x

1  
2 If the conversion of a numeric type results in a loss of precision, no exception is thrown.  
3 However, an exception is thrown if the conversion result is a value that is larger than  
4 that which can be represented by the destination type. For example, when a  
5 `System.Double` is converted to a `System.Single`, a loss of precision might occur but no  
6 exception is thrown. However, if the magnitude of the `System.Double` is too large to be  
7 represented by a `System.Single`, a `System.OverflowException` is thrown.

8

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToBoolean(System.Decimal) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(decimal value)

[C#]
public static bool ToBoolean(decimal value)
```

### Summary

Converts a System.Decimal to a System.Boolean.

### Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### Return Value

true if *value* is non-zero; false if *value* is zero.

### Example

The following example demonstrates converting System.Decimal values to System.Boolean values.

[C#]

```
using System;
class ConvertBoolTest {
    static public void Main() {
        decimal decimal0 = 0m;
        decimal decimal1 = 1m;
        decimal decimal2 = -2m;
        bool bool0 = Convert.ToBoolean(decimal0);
        bool bool1 = Convert.ToBoolean(decimal1);
        bool bool2 = Convert.ToBoolean(decimal2);
        Console.WriteLine("(decimal) {0} as bool = {1}", decimal0, bool0);
        Console.WriteLine("(decimal) {0} as bool = {1}", decimal1, bool1);
        Console.WriteLine("(decimal) {0} as bool = {1}", decimal2, bool2);
    }
}
```

The output is

```
(decimal) 0 as bool = False
```

```
1
2 (decimal) 1 as bool = True
3
4
5 (decimal) -2 as bool = True
6
7
```

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToBoolean(System.Double) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(float64 value)

[C#]
public static bool ToBoolean(double value)
```

### Summary

Converts a System.Double to a System.Boolean.

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

true if *value* is non-zero; false if *value* is zero.

### Example

The following example demonstrates converting System.Double values to System.Boolean values.

```
[C#]
```

```
using System;
class ConvertBoolTest {
    static public void Main() {
        double double0 = 0.0;
        double double1 = 1.0;
        double double2 = -2.0;
        bool bool0 = Convert.ToBoolean(double0);
        bool bool1 = Convert.ToBoolean(double1);
        bool bool2 = Convert.ToBoolean(double2);
        Console.WriteLine("(double) {0} as bool = {1}", double0, bool0);
        Console.WriteLine("(double) {0} as bool = {1}", double1, bool1);
        Console.WriteLine("(double) {0} as bool = {1}", double2, bool2);
    }
}
```

The output is

```
(double) 0 as bool = False
```

```
1
2 (double) 1 as bool = True
3
4
5 (double) -2 as bool = True
6
7
```

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToBoolean(System.Single) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(float32 value)

[C#]
public static bool ToBoolean(float value)
```

### Summary

Converts a System.Single to a System.Boolean.

### Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

### Return Value

true if *value* is non-zero; false if *value* is zero.

### Example

The following example demonstrates converting System.Single values to System.Boolean values.

```
[C#]
```

```
using System;
class ConvertBoolTest {
    static public void Main() {
        float float0 = 0.0f;
        float float1 = 1.0f;
        float float2 = -2.0f;
        bool bool0 = Convert.ToBoolean(float0);
        bool bool1 = Convert.ToBoolean(float1);
        bool bool2 = Convert.ToBoolean(float2);
        Console.WriteLine("(float) {0} as bool = {1}", float0, bool0);
        Console.WriteLine("(float) {0} as bool = {1}", float1, bool1);
        Console.WriteLine("(float) {0} as bool = {1}", float2, bool2);
    }
}
```

The output is

```
(float) 0 as bool = False
```

```
1
2 (float) 1 as bool = True
3
4
5 (float) -2 as bool = True
6
7
```

# 1 Convert.ToBoolean(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static bool ToBoolean(string value)  
4 [C#]  
5 public static bool ToBoolean(string value)
```

## 6 Summary

7 Converts a System.String to a System.Boolean.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted.

## 9 Return Value

11 true if *value* equals System.Boolean.TrueString; false if *value* equals  
12 System.Boolean.FalseString.

## 13 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> is not equal to System.Boolean.TrueString or System.Boolean.FalseString.

## 14 Example

16 The following example demonstrates converting System.String values to  
17 System.Boolean values.

```
18 [C#]  
19  
20 using System;  
21 class ConvertBoolTest {  
22     static public void Main() {  
23         string string0 = Boolean.TrueString;  
24         string string1 = Boolean.FalseString;  
25         string string2 = "foo"; //This is an invalid Boolean.  
26         bool bool0 = Convert.ToBoolean(string0);  
27         bool bool1 = Convert.ToBoolean(string1);
```

```
1         Console.WriteLine("(string) {0} as bool = {1}",string0,bool0);
2         Console.WriteLine("(string) {0} as bool = {1}",string1,bool1);
3         bool bool2 = Convert.ToBoolean(string2); //Throws an exception.
4         Console.WriteLine("(string) {0} as bool = {1}",string2,bool2);
5     }
6 }
7 The output is

8
9 (string) True as bool = True
10 (string) False as bool = False
11 Unhandled Exception: System.FormatException: String was not recognized as a
12 valid Boolean.
13     at System.Boolean.Parse(String value)
14     at Convert.ToBoolean(String value)
15     at ConvertBoolTest.Main() in C:\ECMAExamples\ConvertString.cs:line 12
16
```

# Convert.ToBoolean(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(unsigned int64 value)

[C#]
public static bool ToBoolean(ulong value)
```

## Summary

Converts a System.UInt64 to a System.Boolean.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToBoolean(System.Decimal).

## Example

The following example demonstrates converting System.UInt64 values to System.Boolean values.

```
[C#]
using System;
class ConvertBoolTest {
    static public void Main() {
        ulong ulong0 = 0;
        ulong ulong1 = 1;
        bool bool0 = Convert.ToBoolean(ulong0);
        bool bool1 = Convert.ToBoolean(ulong1);
        Console.WriteLine("(ulong) {0} as bool = {1}",ulong0,bool0);
        Console.WriteLine("(ulong) {0} as bool = {1}",ulong1,bool1);
    }
}
```

```
1 }
2 The output is
3
4 (ulong) 0 as bool = False
5
6
7 (ulong) 1 as bool = True
8
9
```

# Convert.ToBoolean(System.Int64) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(int64 value)

[C#]
public static bool ToBoolean(long value)
```

## Summary

Converts a System.Int64 to a System.Boolean.

## Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Example

The following example demonstrates converting System.Int64 values to System.Boolean values.

```
[C#]
```

```
using System;
class ConvertBoolTest {
    static public void Main() {
        long long0 = 0;
        long long1 = 1;
        long long2 = -2;
        bool bool0 = Convert.ToBoolean(long0);
        bool bool1 = Convert.ToBoolean(long1);
        bool bool2 = Convert.ToBoolean(long2);
        Console.WriteLine("(long) {0} as bool = {1}", long0, bool0);
        Console.WriteLine("(long) {0} as bool = {1}", long1, bool1);
        Console.WriteLine("(long) {0} as bool = {1}", long2, bool2);
    }
}
```

The output is

```
(long) 0 as bool = False
```

```
(long) 1 as bool = True
```

```
1
2
3 (long) -2 as bool = True
4
5
```

# Convert.ToBoolean(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(unsigned int32 value)

[C#]
public static bool ToBoolean(uint value)
```

## Summary

Converts a System.UInt32 to a System.Boolean.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToBoolean(System.Int64).

## Example

The following example demonstrates converting System.UInt32 values to System.Boolean values.

[C#]

```
using System;
class ConvertBoolTest {
    static public void Main() {
        uint uint0 = 0;
        uint uint1 = 1;
        bool bool0 = Convert.ToBoolean(uint0);
        bool bool1 = Convert.ToBoolean(uint1);
        Console.WriteLine("(uint) {0} as bool = {1}",uint0,bool0);
        Console.WriteLine("(uint) {0} as bool = {1}",uint1,bool1);
    }
}
```

```
1 The output is
2
3 (uint) 0 as bool = False
4
5
6 (uint) 1 as bool = True
7
8
```

# Convert.ToBoolean(System.Int32) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(int32 value)

[C#]
public static bool ToBoolean(int value)
```

## Summary

Converts a System.Int32 to a System.Boolean.

## Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Example

The following example demonstrates converting System.Int32 values to System.Boolean values.

```
[C#]
```

```
using System;
class ConvertBoolTest {
    static public void Main() {
        int int0 = 0;
        int int1 = 1;
        int int2 = -2;
        bool bool0 = Convert.ToBoolean(int0);
        bool bool1 = Convert.ToBoolean(int1);
        bool bool2 = Convert.ToBoolean(int2);
        Console.WriteLine("(int) {0} as bool = {1}",int0,bool0);
        Console.WriteLine("(int) {0} as bool = {1}",int1,bool1);
        Console.WriteLine("(int) {0} as bool = {1}",int2,bool2);
    }
}
```

The output is

```
(int) 0 as bool = False
(int) 1 as bool = True
```

```
1
2
3 (int) -2 as bool = True
4
5
```

# Convert.ToBoolean(System.UInt16) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(unsigned int16 value)

[C#]
public static bool ToBoolean(ushort value)
```

## Summary

Converts a System.UInt16 to a System.Boolean.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToBoolean(System.Int32).

## Example

The following example demonstrates converting System.Int16 values to System.Boolean values.

```
[C#]
using System;
class ConvertBoolTest {
    static public void Main() {
        ushort ushort0 = 0;
        ushort ushort1 = 1;
        bool bool0 = Convert.ToBoolean(ushort0);
        bool bool1 = Convert.ToBoolean(ushort1);
        Console.WriteLine("(ushort) {0} as bool = {1}", ushort0, bool0);
        Console.WriteLine("(ushort) {0} as bool = {1}", ushort1, bool1);
    }
}
```

```
1 The output is
2
3 (ushort) 0 as bool = False
4
5
6 (ushort) 1 as bool = True
7
8
```

# Convert.ToBoolean(System.Int16) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(int16 value)

[C#]
public static bool ToBoolean(short value)
```

## Summary

Converts a System.Int16 to a System.Boolean.

## Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Example

The following example demonstrates converting System.Byte values to System.Boolean values.

```
[C#]
```

```
using System;
class ConvertBoolTest {
    static public void Main() {
        short short0 = 0;
        short short1 = 1;
        short short2 = -2;
        bool bool0 = Convert.ToBoolean(short0);
        bool bool1 = Convert.ToBoolean(short1);
        bool bool2 = Convert.ToBoolean(short2);
        Console.WriteLine("(short) {0} as bool = {1}",short0,bool0);
        Console.WriteLine("(short) {0} as bool = {1}",short1,bool1);
        Console.WriteLine("(short) {0} as bool = {1}",short2,bool2);
    }
}
```

The output is

```
(short) 0 as bool = False
(short) 1 as bool = True
```

```
1
2
3 (short) -2 as bool = True
4
5
```

# 1 Convert.ToBoolean(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static bool ToBoolean(unsigned int8 value)  
4 [C#]  
5 public static bool ToBoolean(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.Boolean.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

## 9 Return Value

11 true if *value* is non-zero; false if *value* is zero.

## 12 Example

13 The following example demonstrates converting System.Byte values to System.Boolean  
14 values.

15 [C#]

```
17 using System;  
18 class ConvertBoolTest {  
19     static public void Main() {  
20         byte byte0 = (byte) 0;  
21         byte byte1 = (Byte) 1;  
22         bool bool0 = Convert.ToBoolean(byte0);  
23         bool bool1 = Convert.ToBoolean(byte1);  
24         Console.WriteLine("(byte) {0} as bool = {1}",byte0,bool0);  
25         Console.WriteLine("(byte) {0} as bool = {1}",byte1,bool1);  
26     }  
27 }
```

28 The output is

29  
30 (byte) 0 as bool = False

31  
32  
33 (byte) 1 as bool = True

# Convert.ToBoolean(System.SByte) Method

```
[ILAsm]
.method public hidebysig static bool ToBoolean(int8 value)

[C#]
public static bool ToBoolean(sbyte value)
```

## Summary

Converts a `System.SByte` to a `System.Boolean`.

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

Parameter	Description
<i>value</i>	The 8-bit signed integer value to be converted.

## Return Value

true if *value* is non-zero; false if *value* is zero.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToBoolean(System.Int16)`.

## Example

The following example demonstrates converting `System.SByte` values to `System.Boolean` values.

```
[C#]
using System;
class ConvertBoolTest {
    static public void Main() {
        sbyte sbyte0 = (sbyte) 0;
        sbyte sbyte1 = (sbyte) 1;
        sbyte sbyte2 = (sbyte) -2;
        bool bool0 = Convert.ToBoolean(sbyte0);
        bool bool1 = Convert.ToBoolean(sbyte1);
        bool bool2 = Convert.ToBoolean(sbyte2);
        Console.WriteLine("(sbyte) {0} as bool = {1}", sbyte0, bool0);
        Console.WriteLine("(sbyte) {0} as bool = {1}", sbyte1, bool1);
    }
}
```

```
1         Console.WriteLine("(sbyte) {0} as bool = {1}",sbyte2,bool2);
2     }
3 }
4 The output is
5
6 (sbyte) 0 as bool = False
7
8
9 (sbyte) 1 as bool = True
10
11
12 (sbyte) -2 as bool = True
13
14
```

# 1 Convert.ToBoolean(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static bool ToBoolean(bool value)  
4 [C#]  
5 public static bool ToBoolean(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.Boolean.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

# Convert.ToByte(System.String) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(string value)

[C#]
public static byte ToByte(string value)
```

## Summary

Converts a `System.String` representation of a number to a `System.Byte`.

## Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to be converted. The string is in the <code>System.Globalization.NumberStyles.Integer</code> style.

## Return Value

*value* as a `System.Byte`.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> does not consist of an optional sign followed by one or more digits (zero through nine).
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Byte.MaxValue</code> or less than <code>System.Byte.MinValue</code> .

## Example

The following example demonstrates converting `System.String` values to `System.Byte` values.

```
[C#]
using System;
class ConvertByteTest {
```

```
1     static public void Main() {
2         string string0 = "+22";
3         string string1 = "0";
4         string string2 = "-1";
5         byte byte0 = Convert.ToByte(string0);
6         byte byte1 = Convert.ToByte(string1);
7         Console.WriteLine("(string) {0} as byte = {1}",string0,byte0);
8         Console.WriteLine("(string) {0} as byte = {1}",string1,byte1);
9         byte byte2 = Convert.ToByte(string2);
10        Console.WriteLine("(string) {0} as byte = {1}",string2,byte2);
11    }
12 }
```

13 The output is

14  
15 (string) +22 as byte = 22

16

17

18 (string) 0 as byte = 0

19

20

21 Exception occurred: System.OverflowException: Value was either too large or  
22 too small for an unsigned byte.

23

24

25 at System.Byte.Parse(String s, NumberStyles style, IFormatProvider provider)

26

27

28 at System.Byte.Parse(String s)

29

30

31 at Convert.ToByte(String value)

32

33

34 at ConvertByteTest.Main() in

35 C:\ECMAExamples\ConvertToByte\ConvertString.cs:line 11

36

37

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToByte(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static unsigned int8 ToByte(decimal value)  
6 [C#]  
7 public static byte ToByte(decimal value)
```

### 8 Summary

9 Converts a System.Decimal to a System.Byte.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### 12 Return Value

13 *value* as a System.Byte, rounded to the nearest integer.

### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.  
17 [Note: This process is known as banker's rounding.]  
18  
19

### 20 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue.

### 22 Example

23 The following example demonstrates converting System.Decimal values to System.Byte  
24 values.

```
25 [C#]  
26
```

```

1  using System;
2  class ConvertByteTest {
3      static public void Main() {
4          decimal decimal0 = 0.0m;
5          decimal decimal1 = 1.5m;
6          decimal decimal2 = 2.5m;
7          decimal decimal3 = -1.0m;
8          byte byte0 = Convert.ToByte(decimal0);
9          byte byte1 = Convert.ToByte(decimal1);
10         byte byte2 = Convert.ToByte(decimal2);
11         Console.WriteLine("(decimal) {0} as byte = {1}",decimal0,byte0);
12         Console.WriteLine("(decimal) {0} as byte = {1}",decimal1,byte1);
13         Console.WriteLine("(decimal) {0} as byte = {1}",decimal2,byte2);
14
15         byte byte3 = Convert.ToByte(decimal3); //Throws an exception.
16         Console.WriteLine("(decimal) {0} as byte = {1}",decimal3,byte3);
17     }
18 }
19 The output is
20
21 (decimal) 0 as byte = 0
22
23
24 (decimal) 1.5 as byte = 2
25
26
27 (decimal) 2.5 as byte = 2
28
29
30 Exception occurred: System.OverflowException: Value was either too large or
31 too small for an unsigned byte.
32
33
34 at System.Decimal.ToByte(Decimal value)
35
36
37 at Convert.ToByte(Decimal value)
38
39
40 at ConvertByteTest.Main() in
41 C:\ECMAExamples\ConvertToByte\ConvertDecimal.cs:line 15
42
43

```

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToByte(System.Double) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(float64 value)

[C#]
public static byte ToByte(double value)
```

### Summary

Converts a System.Double to a System.Byte.

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

*value* as a System.Byte, rounded to the nearest integer.

### Description

Prior to the conversion, if *value* is halfway between two numbers, it is rounded to the number that has an even digit in the rightmost decimal position. For example, when rounded to two decimals, the value 2.345 becomes 2.34 and the value 2.355 becomes 2.36

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue, or <i>value</i> is equal to one of System.Double.NaN, System.Double.PositiveInfinity, or System.Double.NegativeInfinity.

### Example

The following example demonstrates converting System.Double values to System.Byte values.

```
[C#]
```

```

1  using System;
2  class ConvertByteTest {
3      static public void Main() {
4          double double0 = 0.0;
5          double double1 = 1.5;
6          double double2 = 2.5;
7          double double3 = -1.0;
8          byte byte0 = Convert.ToByte(double0);
9          byte byte1 = Convert.ToByte(double1);
10         byte byte2 = Convert.ToByte(double2);
11         Console.WriteLine("(double) {0} as byte = {1}",double0,byte0);
12         Console.WriteLine("(double) {0} as byte = {1}",double1,byte1);
13         Console.WriteLine("(double) {0} as byte = {1}",double2,byte2);
14
15         byte byte3 = Convert.ToByte(double3); //Throws an exception.
16         Console.WriteLine("(double) {0} as byte = {1}",double3,byte3);
17     }
18 }
19 The output is
20
21 (double) 0 as byte = 0
22
23
24 (double) 1.5 as byte = 2
25
26
27 (double) 2.5 as byte = 2
28
29
30 Exception occurred: System.OverflowException: Value was either too large or
31 too small for an unsigned byte.
32
33
34 at Convert.ToByte(Int32 value)
35
36
37 at Convert.ToByte(Double value)
38
39
40 at ConvertByteTest.Main() in
41 C:\ECMAExamples\ConvertToByte\ConvertDouble.cs:line 15
42
43

```

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToByte(System.Single) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(float32 value)

[C#]
public static byte ToByte(float value)
```

### Summary

Converts a `System.Single` to a `System.Byte`.

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a `System.Byte`, rounded to the nearest integer.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.Byte.MaxValue</code> or less than <code>System.Byte.MinValue</code> , or <i>value</i> is equal to one of <code>System.Single.NaN</code> , <code>System.Single.PositiveInfinity</code> , or <code>System.Single.NegativeInfinity</code> .

### Example

The following example demonstrates converting `System.Single` values to `System.Byte` values.

```
[C#]
```

```
using System;
```

```

1 class ConvertByteTest {
2     static public void Main() {
3         float float0 = 0.0f;
4         float float1 = 1.5f;
5         float float2 = 2.5f;
6         float float3 = -1.0f;
7         byte byte0 = Convert.ToByte(float0);
8         byte byte1 = Convert.ToByte(float1);
9         byte byte2 = Convert.ToByte(float2);
10        Console.WriteLine("(float) {0} as byte = {1}",float0,byte0);
11        Console.WriteLine("(float) {0} as byte = {1}",float1,byte1);
12        Console.WriteLine("(float) {0} as byte = {1}",float2,byte2);
13
14        byte byte3 = Convert.ToByte(float3); //Throws an exception.
15        Console.WriteLine("(float) {0} as byte = {1}",float3,byte3);
16    }
17 }
18 The output is
19
20 (float) 0 as byte = 0
21
22
23 (float) 1.5 as byte = 2
24
25
26 (float) 2.5 as byte = 2
27
28
29 Exception occurred: System.OverflowException: Value was either too large or
30 too small for an unsigned byte.
31
32
33 at Convert.ToByte(Int32 value)
34
35
36 at Convert.ToByte(Single value)
37
38
39 at ConvertByteTest.Main() in
40 C:\ECMAExamples\ConvertToByte\ConvertFloat.cs:line 15
41
42

```

# 1 Convert.ToByte(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(unsigned int64 value)  
4 [C#]  
5 public static byte ToByte(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.Byte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

## 11 Return Value

12 *value* as a System.Byte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToByte(System.Decimal).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue.

## 18 Example

20 The following example demonstrates converting System.UInt64 values to System.Byte  
21 values.

```
22 [C#]  
23  
24 using System;  
25 class ConvertByteTest {
```

```
1     static public void Main() {
2         ulong ulong0 = 0;
3         ulong ulong1 = 32000;
4         byte byte0 = Convert.ToByte(ulong0);
5         Console.WriteLine("(ulong) {0} as byte = {1}",ulong0,byte0);
6         byte byte1 = Convert.ToByte(ulong1); //Throws an exception.
7         Console.WriteLine("(ulong) {0} as byte = {1}",ulong1,byte1);
8     }
9 }
10 The output is
11
12 (ulong) 0 as byte = 0
13
14
15 Exception occurred: System.OverflowException: Value was either too large or
16 too small for an unsigned byte.
17
18
19 at Convert.ToByte(UInt64 value)
20
21
22 at ConvertByteTest.Main() in
23 C:\ECMAExamples\ConvertToByte\ConvertUInt64.cs:line 8
24
25
```

# 1 Convert.ToInt64(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(int64 value)  
4 [C#]  
5 public static byte ToByte(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.Byte.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a System.Byte.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue.

13

14

# 1 Convert.ToByte(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(unsigned int32 value)  
4 [C#]  
5 public static byte ToByte(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.Byte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

## 11 12 Return Value

13 *value* as a System.Byte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToByte(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue.

# 1 Convert.ToByte(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(int32 value)  
4 [C#]  
5 public static byte ToByte(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.Byte.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

## 9 Return Value

11 *value* as a System.Byte.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue.

13  
14

# Convert.ToByte(System.UInt16) Method

```
[ILAsm]  
.method public hidebysig static unsigned int8 ToByte(unsigned int16 value)  
  
[C#]  
public static byte ToByte(ushort value)
```

## Summary

Converts a System.UInt16 to a System.Byte.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

## Return Value

*value* as a System.Byte.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToByte(System.Int32).

## Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue.

## Example

The following example demonstrates converting System.UInt16 values to System.Byte values.

```
[C#]
```

```
using System;  
class ConvertByteTest {
```

```
1     static public void Main() {
2         ushort ushort0 = 0;
3         ushort ushort1 = 32000;
4         byte byte0 = Convert.ToByte(ushort0);
5         Console.WriteLine("(ushort) {0} as byte = {1}",ushort0,byte0);
6         byte byte1 = Convert.ToByte(ushort1); //Throws an exception.
7         Console.WriteLine("(ushort) {0} as byte = {1}",ushort1,byte1);
8     }
9 }
10 The output is
11
12 (ushort) 0 as byte = 0
13
14
15 Exception occurred: System.OverflowException: Value was either too large or
16 too small for an unsigned byte.
17
18
19 at Convert.ToByte(UInt16 value)
20
21
22 at ConvertByteTest.Main() in
23 C:\ECMAExamples\ConvertToByte\ConvertUInt16.cs:line 8
24
25
```

# 1 Convert.ToInt16(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(int16 value)  
4 [C#]  
5 public static byte ToByte(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.Byte.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a System.Byte.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue.

13

14

# 1 Convert.ToByte(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(int8 value)  
4 [C#]  
5 public static byte ToByte(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.Byte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 8-bit signed integer to be converted.

11

## 12 Return Value

13 *value* as a System.Byte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToByte(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.Byte.MinValue.

18

19

# 1 Convert.ToByte(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(bool value)  
4 [C#]  
5 public static byte ToByte(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.Byte.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 If *value* equals true, returns 1; if *value* equals false, returns 0.

12

# 1 Convert.ToByte(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(unsigned int8 value)  
4 [C#]  
5 public static byte ToByte(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.Byte.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

9

## 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

# 1 Convert.ToByte(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int8 ToByte(valuetype System.Char  
4 value)  
  
5 [C#]  
6 public static byte ToByte(char value)
```

## 7 Summary

8 Converts a System.Char to a System.Byte.

## 9 Parameters

Parameter	Description
<i>value</i>	The Unicode character to be converted interpreted as an unsigned value.

10

## 11 Return Value

12 *value* as a System.Byte.

## 13 Exceptions

Exception	Condition
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Byte.MaxValue.

14

## 15 Example

16 The following example demonstrates converting System.Char values to System.Byte  
17 values.

18

19 [C#]

```
20 using System;  
21 class ConvertByteTest {  
22     static public void Main() {  
23         char char0 = '0';  
24         char char1 = '1';  
25         char char2 = 'a';  
26         byte byte0 = Convert.ToByte(char0);  
27         byte byte1 = Convert.ToByte(char1);  
28         byte byte2 = Convert.ToByte(char2);  
29         Console.WriteLine("(char) {0} as byte = {1}",char0,byte0);
```

```
1         Console.WriteLine("(char) {0} as byte = {1}",char1,byte1);
2         Console.WriteLine("(char) {0} as byte = {1}",char2,byte2);
3     }
4 }
5 The output is
6
7 (char) 0 as byte = 48
8
9
10 (char) 1 as byte = 49
11
12
13 (char) a as byte = 97
14
15
```

# 1 Convert.ToByte(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static unsigned int8 ToByte(string value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static byte ToByte(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.Byte.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 *value* as a System.Byte.

## 14 Description

15 This method parses *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is parsed using the formatting information of the current system  
19 culture.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> does not consist of an optional sign followed by one or more digits (zero through nine).

**System.OverflowException**

The numeric value of *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`.

1

2

# 1 Convert.ToChar(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(string value)  
4 [C#]  
5 public static char ToChar(string value)
```

## 6 Summary

7 Converts a System.String to a System.Char.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The System.String is required to contain a single character.

9

## 10 Return Value

11 *value* as a System.Char.

## 12 Exceptions

Exception	Condition
<b>System.FormatException</b>	<i>value</i> does not contain exactly one character.

13

14

# 1 Convert.ToChar(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(unsigned  
4 int64 value)  
  
5 [C#]  
6 public static char ToChar(ulong value)
```

## 7 Summary

8 Converts a System.UInt64 to a System.Char.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a System.Char.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToChar(System.Int64).

## 18 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Char.MaxValue.

19

20

# 1 Convert.ToIntChar(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(int64 value)  
4 [C#]  
5 public static char ToChar(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.Char.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a System.Char.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Char.MaxValue or less than System.Char.MinValue.

13

14

# 1 Convert.ToChar(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(unsigned  
4 int32 value)  
  
5 [C#]  
6 public static char ToChar(uint value)
```

## 7 Summary

8 Converts a System.UInt32 to a System.Char.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a System.Char.

## 15 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Char.MaxValue.

16

17

# 1 Convert.ToInt32(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(int32 value)  
4 [C#]  
5 public static char ToChar(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.Char.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a System.Char.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Char.MaxValue or less than System.Char.MinValue.

13

14

# 1 Convert.ToChar(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(unsigned  
4 int16 value)  
  
5 [C#]  
6 public static char ToChar(ushort value)
```

## 7 Summary

8 Converts a System.UInt16 to a System.Char.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a System.Char.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToChar(System.Int32).

18

# 1 Convert.ToInt16(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(int16 value)  
4 [C#]  
5 public static char ToChar(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.Char.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a System.Char.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.Char.MinValue.

13

14

# 1 Convert.ToChar(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(unsigned int8  
4 value)  
5 [C#]  
6 public static char ToChar(byte value)
```

## 7 Summary

8 Converts a System.Byte to a System.Char.

## 9 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

10

## 11 Return Value

12 *value* as a System.Char.

13

# 1 Convert.ToChar(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(int8 value)  
4 [C#]  
5 public static char ToChar(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.Char.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a System.Char.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToChar(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	value is less than System.Char.MinValue.

18

19

# 1 Convert.ToChar(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.Char ToChar(valuetype  
4 System.Char value)  
5 [C#]  
6 public static char ToChar(char value)
```

## 7 Summary

8 Converts a System.Char to a System.Char.

## 9 Parameters

Parameter	Description
<i>value</i>	The Unicode character to be converted.

10

## 11 Return Value

12 *value* is returned unchanged.

## 13 Description

14 [Note: This method is provided for completeness.]  
15  
16

17

# 1 Convert.ToDateTime(System.DateTime)

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig static valuetype System.DateTime  
5 ToDateTime(valuetype System.DateTime value)  
  
6 [C#]  
7 public static DateTime ToDateTime(DateTime value)
```

### 8 Summary

9 Converts a System.DateTime to a System.DateTime.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.DateTime to be converted.

11

### 12 Return Value

13 *value* is returned unchanged.

### 14 Description

15 [Note: This method is provided for completeness.]

16

17

18

# 1 Convert.ToDateTime(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static valuetype System.DateTime  
4 ToDateTime(string value)  
  
5 [C#]  
6 public static DateTime ToDateTime(string value)
```

## 7 Summary

8 Converts a System.String to a System.DateTime structure.

## 9 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in a form allowed by the System.DateTime.Parse(System.String) method.

10

## 11 Return Value

12 *value* as a System.DateTime.

## 13 Description

14 This method parses *value* using the information in a  
15 System.Globalization.DateTimeFormatInfo instance initialized for the current system  
16 culture.

## 17 Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a System.DateTime.

18

19

# 1 Convert.ToDateTime(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static valuetype System.DateTime  
5 ToDateTime(string value, class System.IFormatProvider provider)  
  
6 [C#]  
7 public static DateTime ToDateTime(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.DateTime structure.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.DateTimeFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 *value* as a System.DateTime.

## 14 Description

15 This method parses *value* using the information in the  
16 System.Globalization.DateTimeFormatInfo instance supplied by *provider*. If *provider*  
17 is null or if a System.Globalization.DateTimeFormatInfo cannot be obtained from  
18 *provider*, the string is parsed using the formatting information of the current system  
19 culture.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a System.DateTime.

21

22

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Boolean) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(bool value)  
  
6 [C#]  
7 public static decimal ToDecimal(bool value)
```

#### 8 Summary

9 Converts a System.Boolean to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

#### 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(decimal value)  
6 [C#]  
7 public static decimal ToDecimal(decimal value)
```

#### 8 Summary

9 Converts a System.Decimal to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

11

#### 12 Return Value

13 *value* is returned unchanged.

#### 14 Description

15 [Note: This method is provided for completeness.]  
16  
17

18

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToDecimal(System.String, System.IFormatProvider) Method

```
[ILAsm]  
.method public hidebysig static decimal ToDecimal(string value, class  
System.IFormatProvider provider)  
  
[C#]  
public static decimal ToDecimal(string value, IFormatProvider provider)
```

### Summary

Converts a System.String to a System.Decimal.

### Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Number style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### Return Value

*value* as a System.Decimal.

### Description

This method parses *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is parsed using the formatting information of the current system culture.

### Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is a null reference.

<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Decimal.MaxValue</code> or less than <code>System.Decimal.MinValue</code> .

1

2

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToDecimal(System.String) Method

```
[ILAsm]
.method public hidebysig static decimal ToDecimal(string value)

[C#]
public static decimal ToDecimal(string value)
```

### Summary

Converts a System.String to a System.Decimal.

### Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Number style.

### Return Value

*value* as a System.Decimal.

### Description

This method parses *value* using the information in a System.Globalization.NumberFormatInfo instance initialized for the current system culture.

### Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Decimal.MaxValue or less than System.Decimal.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(float64 value)  
  
6 [C#]  
7 public static decimal ToDecimal(double value)
```

### 8 Summary

9 Converts a System.Double to a System.Decimal.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### 12 Return Value

13 *value* as a System.Decimal. The System.Decimal contains 15 significant digits and is  
14 rounded using banker's rounding.

### 15 Exceptions

Exception	Condition
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Decimal.MaxValue or less than System.Decimal.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(float32 value)  
  
6 [C#]  
7 public static decimal ToDecimal(float value)
```

### 8 Summary

9 Converts a System.Single to a System.Decimal.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

### 12 Return Value

13 *value* as a System.Decimal. The System.Decimal contains 7 significant digits and is  
14 rounded using banker's rounding.

### 15 Exceptions

Exception	Condition
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Decimal.MaxValue or less than System.Decimal.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.UInt64) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(unsigned int64 value)  
  
6 [C#]  
7 public static decimal ToDecimal(ulong value)
```

### 8 Summary

9 Converts a System.UInt64 to a System.Decimal.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

### 13 Return Value

14 *value* as a System.Decimal.

### 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Decimal).  
18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Int64) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(int64 value)  
6 [C#]  
7 public static decimal ToDecimal(long value)
```

#### 8 Summary

9 Converts a System.Int64 to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

#### 12 Return Value

13 *value* as a System.Decimal.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.UInt32) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(unsigned int32 value)  
6 [C#]  
7 public static decimal ToDecimal(uint value)
```

### 8 Summary

9 Converts a System.UInt32 to a System.Decimal.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

### 13 Return Value

14 *value* as a System.Decimal.

### 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Int64).  
18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(int32 value)  
6 [C#]  
7 public static decimal ToDecimal(int value)
```

#### 8 Summary

9 Converts a System.Int32 to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

#### 12 Return Value

13 *value* as a System.Decimal.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.UInt16) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(unsigned int16 value)  
6 [C#]  
7 public static decimal ToDecimal(ushort value)
```

### 8 Summary

9 Converts a System.UInt16 to a System.Decimal.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

### 13 Return Value

14 *value* as a System.Decimal.

### 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Int32).  
18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Int16) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(int16 value)  
6 [C#]  
7 public static decimal ToDecimal(short value)
```

#### 8 Summary

9 Converts a System.Int16 to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

#### 12 Return Value

13 *value* as a System.Decimal.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDecimal(System.Byte) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(unsigned int8 value)  
6 [C#]  
7 public static decimal ToDecimal(byte value)
```

#### 8 Summary

9 Converts a System.Byte to a System.Decimal.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

11

#### 12 Return Value

13 *value* as a System.Decimal.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDecimal(System.SByte) Method

```
4 [ILAsm]  
5 .method public hidebysig static decimal ToDecimal(int8 value)  
6 [C#]  
7 public static decimal ToDecimal(sbyte value)
```

### 8 Summary

9 Converts a System.SByte to a System.Decimal.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

### 14 Return Value

15 *value* as a System.Decimal.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToDecimal(System.Int16).

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(float64 value)  
6 [C#]  
7 public static double ToDouble(double value)
```

#### 8 Summary

9 Converts a System.Double to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

11

#### 12 Return Value

13 *value* is returned unchanged.

#### 14 Description

15 [Note: This method is provided for completeness.]  
16  
17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(float32 value)  
6 [C#]  
7 public static double ToDouble(float value)
```

#### 8 Summary

9 Converts a System.Single to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

11

#### 12 Return Value

13 *value* as a System.Double.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.UInt64) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(unsigned int64 value)  
6 [C#]  
7 public static double ToDouble(ulong value)
```

### 8 Summary

9 Converts a System.UInt64 to a System.Double.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

13

### 14 Return Value

15 *value* as a System.Double.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToDouble(System.Decimal).

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.Int64) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(int64 value)  
6 [C#]  
7 public static double ToDouble(long value)
```

### 8 Summary

9 Converts a System.Int64 to a System.Double.

### 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

### 12 Return Value

13 *value* as a System.Double.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.UInt32) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(unsigned int32 value)  
6 [C#]  
7 public static double ToDouble(uint value)
```

### 8 Summary

9 Converts a System.UInt32 to a System.Double.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

13

### 14 Return Value

15 *value* as a System.Double.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToDouble(System.Int64).

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(int32 value)  
6 [C#]  
7 public static double ToDouble(int value)
```

#### 8 Summary

9 Converts a System.Int32 to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

#### 12 Return Value

13 *value* as a System.Double.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.UInt16) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(unsigned int16 value)  
6 [C#]  
7 public static double ToDouble(ushort value)
```

### 8 Summary

9 Converts a System.UInt16 to a System.Double.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

### 13 Return Value

14 *value* as a System.Double.

### 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDouble(System.Int32).  
18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.Int16) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(int16 value)  
6 [C#]  
7 public static double ToDouble(short value)
```

### 8 Summary

9 Converts a System.Int16 to a System.Double.

### 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

### 12 Return Value

13 *value* as a System.Double.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Byte) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(unsigned int8 value)  
6 [C#]  
7 public static double ToDouble(byte value)
```

#### 8 Summary

9 Converts a System.Byte to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

11

#### 12 Return Value

13 *value* as a System.Double.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.SByte) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(int8 value)  
6 [C#]  
7 public static double ToDouble(sbyte value)
```

### 8 Summary

9 Converts a System.SByte to a System.Double.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

13

### 14 Return Value

15 *value* as a System.Double.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToDouble(System.Int16).

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Boolean) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(bool value)  
6 [C#]  
7 public static double ToDouble(bool value)
```

#### 8 Summary

9 Converts a System.Boolean to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

#### 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToDouble(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static float64 ToDouble(decimal value)  
6 [C#]  
7 public static double ToDouble(decimal value)
```

#### 8 Summary

9 Converts a System.Decimal to a System.Double.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

11

#### 12 Return Value

13 *value* as a System.Double. *value* is rounded using banker's rounding.

14

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToDouble(System.String) Method

```
[ILAsm]
.method public hidebysig static float64 ToDouble(string value)

[C#]
public static double ToDouble(string value)
```

### Summary

Converts a System.String to a System.Double.

### Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Float   System.Globalization.NumberStyles.AllowThousands style.

### Return Value

*value* as a System.Double.

### Description

This method parses *value* using the information in a System.Globalization.NumberFormatInfo instance initialized for the current system culture.

### Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Double.MaxValue or less than System.Double.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToDouble(System.String, 4 System.IFormatProvider) Method

```
5 [ILAsm]  
6 .method public hidebysig static float64 ToDouble(string value, class  
7 System.IFormatProvider provider)  
  
8 [C#]  
9 public static double ToDouble(string value, IFormatProvider provider)
```

### 10 Summary

11 Converts a System.String to a System.Double.

### 12 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Float   System.Globalization.NumberStyles.AllowThousands style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### 14 Return Value

15 *value* as a System.Double.

### 16 Description

17 This method parses *value* using the information in the  
18 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
19 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
20 *provider*, the string is parsed using the formatting information of the current system  
21 culture.

### 22 Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>value</i> is a null reference.

<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Double.MaxValue</code> or less than <code>System.Double.MinValue</code> .

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt16(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static int16 ToInt16(decimal value)  
  
6 [C#]  
7 public static short ToInt16(decimal value)
```

### 8 Summary

9 Converts a System.Decimal to a System.Int16.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### 12 Return Value

13 *value* as a 16-bit signed integer. *value* is rounded prior to conversion.

### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue or less than System.Int16.MinValue.

# 1 Convert.ToInt16(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(string value)  
4 [C#]  
5 public static short ToInt16(string value)
```

## 6 Summary

7 Converts a System.String to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.

9

## 10 Return Value

11 *value* as a 16-bit signed integer.

## 12 Description

13 This method parses *value* using the information in a  
14 System.Globalization.NumberFormatInfo instance initialized for the current system  
15 culture.

## 16 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Int16.MaxValue or less than System.Int16.MinValue.

17

18

# 1 Convert.ToInt16(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static int16 ToInt16(string value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static short ToInt16(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.Int16.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## 11 12 Return Value

13 *value* as a 16-bit signed integer.

## 14 Description

15 This method parses *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is parsed using the formatting information of the current system  
19 culture.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.

**System.OverflowException**

The numeric value of *value* is greater than `System.Int16.MaxValue` or less than `System.Int16.MinValue`.

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt16(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static int16 ToInt16(float64 value)  
  
6 [C#]  
7 public static short ToInt16(double value)
```

### 8 Summary

9 Converts a System.Double to a System.Int16.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### 12 Return Value

13 *value* as a 16-bit signed integer. *value* is rounded prior to conversion.

### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue or less than System.Int16.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt16(System.Single) Method

```
[ILAsm]
.method public hidebysig static int16 ToInt16(float32 value)

[C#]
public static short ToInt16(float value)
```

### Summary

Converts a `System.Single` to a `System.Int16`.

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a 16-bit signed integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.Int16.MaxValue</code> or less than <code>System.Int16.MinValue</code> .

# 1 Convert.ToInt16(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(unsigned int64 value)  
4 [C#]  
5 public static short ToInt16(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.Int16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

## 11 12 Return Value

13 *value* as a 16-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Decimal).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue.

18  
19

# 1 Convert.ToInt16(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(int64 value)  
4 [C#]  
5 public static short ToInt16(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 16-bit signed integer.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	value is greater than System.Int16.MaxValue or less than System.Int16.MinValue.

13

14

# 1 Convert.ToInt16(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(int16 value)  
4 [C#]  
5 public static short ToInt16(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

# 1 Convert.ToInt16(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(unsigned int32 value)  
4 [C#]  
5 public static short ToInt16(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.Int16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue.

18

19

# 1 Convert.ToInt16(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(int32 value)  
4 [C#]  
5 public static short ToInt16(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 16-bit signed integer.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue or less than System.Int16.MinValue.

13

14

# 1 Convert.ToInt16(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(unsigned int16 value)  
4 [C#]  
5 public static short ToInt16(ushort value)
```

## 6 Summary

7 Converts a System.UInt16 to a System.Int16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int32).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int16.MaxValue.

18

19

# 1 Convert.ToInt16(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(unsigned int8 value)  
4 [C#]  
5 public static short ToInt16(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

9

## 10 Return Value

11 *value* as a 16-bit signed integer.

12

# 1 Convert.ToInt16(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(int8 value)  
4 [C#]  
5 public static short ToInt16(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.Int16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int16).

17

# 1 Convert.ToInt16(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(valuetype System.Char value)  
4 [C#]  
5 public static short ToInt16(char value)
```

## 6 Summary

7 Converts a System.Char to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

9

## 10 Return Value

11 *value* as a 16-bit signed integer.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Int16.MaxValue.

13

14

# 1 Convert.ToInt16(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static int16 ToInt16(bool value)  
4 [C#]  
5 public static short ToInt16(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.Int16.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 If *value* is true returns 1; if *value* is false returns 0.

12

# 1 Convert.ToInt32(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(bool value)  
4 [C#]  
5 public static int ToInt32(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 If *value* is true returns 1; if *value* is false returns 0.

12

# 1 Convert.ToInt32(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static int32 ToInt32(string value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static int ToInt32(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.Int32.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 *value* as a 32-bit signed integer.

## 14 Description

15 This method parses *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is parsed using the formatting information of the current system  
19 culture.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.

**System.OverflowException**

The numeric value of *value* is greater than `System.Int32.MaxValue` or less than `System.Int32.MinValue`.

1

2

# Convert.ToInt32(System.String) Method

```
[ILAsm]  
.method public hidebysig static int32 ToInt32(string value)  
  
[C#]  
public static int ToInt32(string value)
```

## Summary

Converts a `System.String` to a `System.Int32`.

## Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to be converted. The string is in the <code>System.Globalization.NumberStyles.Integer</code> style.

## Return Value

*value* as a 32-bit signed integer.

## Description

This method parses *value* using the information in a `System.Globalization.NumberFormatInfo` instance initialized for the current system culture.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Int32.MaxValue</code> or less than <code>System.Int32.MinValue</code> .

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt32(System.Decimal) Method

```
[ILAsm]  
.method public hidebysig static int32 ToInt32(decimal value)  
  
[C#]  
public static int ToInt32(decimal value)
```

### Summary

Converts a System.Decimal to a System.Int32.

### Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### Return Value

*value* as a 32-bit signed integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int32.MaxValue or less than System.Int32.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt32(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static int32 ToInt32(float64 value)  
  
6 [C#]  
7 public static int ToInt32(double value)
```

### 8 Summary

9 Converts a System.Double to a System.Int32.

### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### 12 Return Value

13 *value* as a 32-bit signed integer. *value* is rounded prior to conversion.

### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int32.MaxValue or less than System.Int32.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt32(System.Single) Method

```
[ILAsm]  
.method public hidebysig static int32 ToInt32(float32 value)  
  
[C#]  
public static int ToInt32(float value)
```

### Summary

Converts a `System.Single` to a `System.Int32`.

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a 32-bit signed integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.Int32.MaxValue</code> or less than <code>System.Int32.MinValue</code> .

# 1 Convert.ToInt32(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(unsigned int64 value)  
4 [C#]  
5 public static int ToInt32(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.Int32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Decimal).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int32.MaxValue.

18

19

# 1 Convert.ToInt32(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(int64 value)  
4 [C#]  
5 public static int ToInt32(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 32-bit signed integer.

## 12 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int32.MaxValue or less than System.Int32.MinValue.

13

14

# 1 Convert.ToInt32(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(int32 value)  
4 [C#]  
5 public static int ToInt32(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

## 9 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

# 1 Convert.ToInt32(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(unsigned int32 value)  
4 [C#]  
5 public static int ToInt32(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.Int32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int32.MaxValue.

18

19

# 1 Convert.ToInt32(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(unsigned int16 value)  
4 [C#]  
5 public static int ToInt32(ushort value)
```

## 6 Summary

7 Converts a System.UInt16 to a System.Int32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int32).

17

# 1 Convert.ToInt32(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(int16 value)  
4 [C#]  
5 public static int ToInt32(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 32-bit signed integer.

12

# 1 Convert.ToInt32(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(unsigned int8 value)  
4 [C#]  
5 public static int ToInt32(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

## 9 Return Value

10 *value* as a 32-bit signed integer.

12

# 1 Convert.ToInt32(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(int8 value)  
4 [C#]  
5 public static int ToInt32(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.Int32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int16).

17

# 1 Convert.ToInt32(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static int32 ToInt32(valuetype System.Char value)  
4 [C#]  
5 public static int ToInt32(char value)
```

## 6 Summary

7 Converts a System.Char to a System.Int32.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

9

## 10 Return Value

11 *value* as a 32-bit signed integer.

12

# 1 Convert.ToInt64(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(valuetype System.Char value)  
4 [C#]  
5 public static long ToInt64(char value)
```

## 6 Summary

7 Converts a System.Char to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

9

## 10 Return Value

11 *value* as a 64-bit signed integer.

12

# 1 Convert.ToInt64(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(bool value)  
4 [C#]  
5 public static long ToInt64(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 If *value* is true returns 1; if *value* is false returns 0.

12

# 1 Convert.ToInt64(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(int8 value)  
4 [C#]  
5 public static long ToInt64(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.Int64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int16).

17

# 1 Convert.ToInt64(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(unsigned int8 value)  
4 [C#]  
5 public static long ToInt64(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

9

## 10 Return Value

11 *value* as a 64-bit signed integer.

12

# 1 Convert.ToInt64(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(int16 value)  
4 [C#]  
5 public static long ToInt64(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 64-bit signed integer.

12

# 1 Convert.ToInt64(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(unsigned int16 value)  
4 [C#]  
5 public static long ToInt64(ushort value)
```

## 6 Summary

7 Converts a System.UInt16 to a System.Int64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int32).

17

# 1 Convert.ToInt64(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(int32 value)  
4 [C#]  
5 public static long ToInt64(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* as a 64-bit signed integer.

12

# 1 Convert.ToInt64(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(unsigned int32 value)  
4 [C#]  
5 public static long ToInt64(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.Int64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int64).

17

# 1 Convert.ToInt64(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(unsigned int64 value)  
4 [C#]  
5 public static long ToInt64(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.Int64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit signed integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Decimal).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int64.MaxValue.

18

19

# 1 Convert.ToInt64(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int64 ToInt64(int64 value)  
4 [C#]  
5 public static long ToInt64(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.Int64.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToInt64(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static int64 ToInt64(float32 value)  
6 [C#]  
7 public static long ToInt64(float value)
```

#### 8 Summary

9 Converts a System.Single to a System.Int64.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

#### 12 Return Value

13 *value* as a 64-bit signed integer. *value* is rounded prior to conversion.

#### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

#### 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int64.MaxValue or less than System.Int64.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt64(System.Double) Method

```
[ILAsm]
.method public hidebysig static int64 ToInt64(float64 value)

[C#]
public static long ToInt64(double value)
```

### Summary

Converts a System.Double to a System.Int64.

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

*value* as a 64-bit signed integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int64.MaxValue or less than System.Int64.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToInt64(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static int64 ToInt64(decimal value)  
  
6 [C#]  
7 public static long ToInt64(decimal value)
```

#### 8 Summary

9 Converts a System.Decimal to a System.Int64.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

#### 12 Return Value

13 *value* as a 64-bit signed integer. *value* is rounded prior to conversion.

#### 14 Description

15 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
16 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

#### 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Int64.MaxValue or less than System.Int64.MinValue.

# Convert.ToInt64(System.String) Method

```
[ILAsm]  
.method public hidebysig static int64 ToInt64(string value)  
  
[C#]  
public static long ToInt64(string value)
```

## Summary

Converts a `System.String` to a `System.Int64`.

## Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to be converted. The string is in the <code>System.Globalization.NumberStyles.Integer</code> style.

## Return Value

*value* as a 64-bit signed integer.

## Description

This method parses *value* using the information in a `System.Globalization.NumberFormatInfo` instance initialized for the current system culture.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Int64.MaxValue</code> or less than <code>System.Int64.MinValue</code> .

# 1 Convert.ToInt64(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static int64 ToInt64(string value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static long ToInt64(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.Int64.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## 11 12 Return Value

13 *value* as a 64-bit signed integer.

## 14 Description

15 This method parses *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is parsed using the formatting information of the current system  
19 culture.

## 20 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.

**System.OverflowException**

The numeric value of *value* is greater than `System.Int64.MaxValue` or less than `System.Int64.MinValue`.

1

2

# 1 Convert.ToSByte(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(bool value)  
4 [C#]  
5 public static sbyte ToSByte(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

## 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Boolean).

17

# 1 Convert.ToSByte(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(int8 value)  
4 [C#]  
5 public static sbyte ToSByte(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* is returned unchanged.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int16).

17

18 [Note: This method is provided for completeness.]

19

20

21

# 1 Convert.ToSByte(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(valuetype System.Char value)  
4 [C#]  
5 public static sbyte ToSByte(char value)
```

## 6 Summary

7 Converts a System.Char to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Char).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.SByte.MaxValue.

18

19

# 1 Convert.ToSByte(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(unsigned int8 value)  
4 [C#]  
5 public static sbyte ToSByte(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Byte).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue.

18

19

# 1 Convert.ToInt16(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(int16 value)  
4 [C#]  
5 public static sbyte ToSByte(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue.

18

19

# Convert.ToSByte(System.UInt16) Method

```
[ILAsm]  
.method public hidebysig static int8 ToSByte(unsigned int16 value)  
  
[C#]  
public static sbyte ToSByte(ushort value)
```

## Summary

Converts a System.UInt16 to a System.SByte.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

## Return Value

*value* as a System.SByte.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt16(System.Int32).

## Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue.

# Convert.ToInt32(System.Int32) Method

```
[ILAsm]  
.method public hidebysig static int8 ToSByte(int32 value)  
  
[C#]  
public static sbyte ToSByte(int value)
```

## Summary

Converts a `System.Int32` to a `System.SByte`.

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

## Return Value

*value* as a `System.SByte`.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt16(System.Int32)`.

## Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> .

# 1 Convert.ToSByte(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(unsigned int32 value)  
4 [C#]  
5 public static sbyte ToSByte(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue.

18

19

# 1 Convert.ToInt64(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(int64 value)  
4 [C#]  
5 public static sbyte ToSByte(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue.

18

19

# 1 Convert.ToSByte(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static int8 ToSByte(unsigned int64 value)  
4 [C#]  
5 public static sbyte ToSByte(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.SByte.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

11

## 12 Return Value

13 *value* as a System.SByte.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt16(System.Decimal).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue.

18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToSByte(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static int8 ToSByte(float32 value)  
  
6 [C#]  
7 public static sbyte ToSByte(float value)
```

### 8 Summary

9 Converts a System.Single to a System.SByte.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

### 14 Return Value

15 *value* as a System.SByte, rounded to the nearest integer.

### 16 Description

17 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
18 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

19  
20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToInt16(System.Single).

### 22 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToSByte(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static int8 ToSByte(float64 value)  
  
6 [C#]  
7 public static sbyte ToSByte(double value)
```

### 8 Summary

9 Converts a System.Double to a System.SByte.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### 14 Return Value

15 *value* as a System.SByte, rounded to the nearest integer.

### 16 Description

17 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
18 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

19  
20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToInt16(System.Double).

### 22 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToSByte(System.Decimal) Method

```
[ILAsm]
.method public hidebysig static int8 ToSByte(decimal value)

[C#]
public static sbyte ToSByte(decimal value)
```

### Summary

Converts a `System.Decimal` to a `System.SByte`.

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Decimal</code> value to be converted.

### Return Value

*value* as a `System.SByte`, rounded to the nearest integer.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt16(System.Decimal)`.

### Exceptions

Exception	Condition
<code>System.OverflowException</code>	<i>value</i> is greater than <code>System.SByte.MaxValue</code> or less than <code>System.SByte.MinValue</code> .

# Convert.ToSByte(System.String) Method

```
[ILAsm]  
.method public hidebysig static int8 ToSByte(string value)  
  
[C#]  
public static sbyte ToSByte(string value)
```

## Summary

Converts a `System.String` representation of a number to a `System.SByte`.

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to be converted. The string is in the <code>System.Globalization.NumberStyles.Integer</code> style.

## Return Value

*value* as a `System.SByte`.

## Description

This method parses *value* using the information in a `System.Globalization.NumberFormatInfo` instance initialized for the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt16(System.String)`.

## Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>value</i> is a null reference.
<code>System.FormatException</code>	<i>value</i> cannot be converted to a numeric value in the specified format.

**System.OverflowException**

The numeric value of *value* is greater than `System.SByte.MaxValue` or less than `System.SByte.MinValue`.

1

2

# Convert.ToSByte(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static int8 ToSByte(string value, class
System.IFormatProvider provider)

[C#]
public static sbyte ToSByte(string value, IFormatProvider provider)
```

## Summary

Converts a System.String to a System.SByte.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## Return Value

*value* as a System.SByte.

## Description

This method parses *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is parsed using the formatting information of the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt16(System.String, System.IFormatProvider).

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToSingle(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static float32 ToSingle(string value, class
System.IFormatProvider provider)

[C#]
public static float ToSingle(string value, IFormatProvider provider)
```

### Summary

Converts a System.String to a System.Single.

### Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Float   System.Globalization.NumberStyles.AllowThousands style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### Return Value

*value* as a System.Single.

### Description

This method parses *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is parsed using the formatting information of the current system culture.

### Exceptions

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

<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.Single.MaxValue</code> or less than <code>System.Single.MinValue</code> .

1

2

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToSingle(System.String) Method

```
[ILAsm]
.method public hidebysig static float32 ToSingle(string value)

[C#]
public static float ToSingle(string value)
```

### Summary

Converts a System.String to a System.Single.

### Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Float   System.Globalization.NumberStyles.AllowThousands style.

### Return Value

*value* as a System.Single.

### Description

This method parses *value* using the information in a System.Globalization.NumberFormatInfo instance initialized for the current system culture.

### Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.Single.MaxValue or less than System.Single.MinValue.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToSingle(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(decimal value)  
  
6 [C#]  
7 public static float ToSingle(decimal value)
```

#### 8 Summary

9 Converts a System.Decimal to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

11

#### 12 Return Value

13 *value* as a System.Single. *value* is rounded using banker's rounding.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToSingle(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(float64 value)  
6 [C#]  
7 public static float ToSingle(double value)
```

#### 8 Summary

9 Converts a System.Double to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

11

#### 12 Return Value

13 *value* as a System.Single.

#### 14 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.Single.MaxValue or less than System.Single.MinValue.

15

16

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToSingle(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(float32 value)  
  
6 [C#]  
7 public static float ToSingle(float value)
```

#### 8 Summary

9 Converts a System.Single to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

11

#### 12 Return Value

13 *value* as a System.Single.

#### 14 Description

15 [Note: This method is provided for completeness.]  
16  
17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToSingle(System.UInt64) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(unsigned int64 value)  
6 [C#]  
7 public static float ToSingle(ulong value)
```

### 8 Summary

9 Converts a System.UInt64 to a System.Single.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

13

### 14 Return Value

15 *value* as a System.Single.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToSingle(System.Decimal).

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToIntSingle(System.Int64) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(int64 value)  
6 [C#]  
7 public static float ToSingle(long value)
```

### 8 Summary

9 Converts a System.Int64 to a System.Single.

### 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

### 12 Return Value

13 *value* as a System.Single.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt32(System.UInt32) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(unsigned int32 value)  
6 [C#]  
7 public static float ToSingle(uint value)
```

### 8 Summary

9 Converts a System.UInt32 to a System.Single.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

13

### 14 Return Value

15 *value* as a System.Single.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToInt32(System.Int64).

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToInt32(System.Int32) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(int32 value)  
6 [C#]  
7 public static float ToSingle(int value)
```

#### 8 Summary

9 Converts a System.Int32 to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

#### 12 Return Value

13 *value* as a System.Single.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt16(System.UInt16) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(unsigned int16 value)  
6 [C#]  
7 public static float ToSingle(ushort value)
```

### 8 Summary

9 Converts a System.UInt16 to a System.Single.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

### 13 Return Value

14 *value* as a System.Single.

### 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt16(System.Int32).  
18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToIntSingle(System.Int16) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(int16 value)  
6 [C#]  
7 public static float ToSingle(short value)
```

### 8 Summary

9 Converts a System.Int16 to a System.Single.

### 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

### 12 Return Value

13 *value* as a System.Single.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToSingle(System.Byte) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(unsigned int8 value)  
6 [C#]  
7 public static float ToSingle(byte value)
```

#### 8 Summary

9 Converts a System.Byte to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

11

#### 12 Return Value

13 *value* as a System.Single.

14

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToSingle(System.SByte) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(int8 value)  
6 [C#]  
7 public static float ToSingle(sbyte value)
```

### 8 Summary

9 Converts a System.SByte to a System.Single.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

### 14 Return Value

15 *value* as a System.Single.

### 16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative, use  
18 System.Convert.ToSingle(System.Int16).

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToSingle(System.Boolean) Method

```
4 [ILAsm]  
5 .method public hidebysig static float32 ToSingle(bool value)  
6 [C#]  
7 public static float ToSingle(bool value)
```

#### 8 Summary

9 Converts a System.Boolean to a System.Single.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

#### 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

14

# Convert.ToString(System.UInt32, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(unsigned int32 value,
class System.IFormatProvider provider)

[C#]
public static string ToString(uint value, IFormatProvider provider)
```

## Summary

Converts a System.UInt32 to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## Return Value

Returns the value returned by *value*.ToString(*provider*).

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToString(System.Int64, System.IFormatProvider).

This method converts *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is formatted in accordance with the current system culture.

[Note: See System.UInt32.ToString.]

# 1 Convert.ToString(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(int64 value)  
4 [C#]  
5 public static string ToString(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.String.

## 8 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

9

## 10 Return Value

11 Returns the value returned by *value*.ToString().

## 12 Description

13 [Note: See System.Int64.ToString.]  
14  
15

16

# 1 Convert.ToString(System.Int64, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static string ToString(int64 value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static string ToString(long value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.Int64 to a System.String.

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString(*provider*).

## 14 Description

15 This method converts *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is formatted in accordance with the current system culture.

19

20 [Note: See System.Int64.ToString.]

21

22

23

# 1 Convert.ToString(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(unsigned int64 value)  
4 [C#]  
5 public static string ToString(ulong value)
```

## 6 Summary

7 Converts a System.UInt64 to a System.String.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString().

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToString(System.Decimal).

17

18 [*Note:* See System.UInt64.ToString.]

19

20

21

# Convert.ToString(System.UInt64, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(unsigned int64 value,
class System.IFormatProvider provider)

[C#]
public static string ToString(ulong value, IFormatProvider provider)
```

## Summary

Converts a System.UInt64 to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## Return Value

Returns the value returned by *value*.ToString(*provider*).

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToString(System.Decimal, System.IFormatProvider).

This method converts *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is formatted in accordance with the current system culture.

[Note: See System.UInt64.ToString.]

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToString(System.Single) Method

```
4 [ILAsm]  
5 .method public hidebysig static string ToString(float32 value)  
6 [C#]  
7 public static string ToString(float value)
```

#### 8 Summary

9 Converts a System.Single to a System.String.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.

11

#### 12 Return Value

13 Returns the value returned by *value*.ToString().

#### 14 Description

15 [Note: See System.Single.ToString.]  
16  
17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToString(System.Single, 4 System.IFormatProvider) Method

```
5 [ILAsm]  
6 .method public hidebysig static string ToString(float32 value, class  
7 System.IFormatProvider provider)  
8 [C#]  
9 public static string ToString(float value, IFormatProvider provider)
```

### 10 Summary

11 Converts a System.Single to a System.String.

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Single value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### 14 Return Value

15 Returns the value returned by *value*.ToString(*provider*).

### 16 Description

17 This method converts *value* using the information in the  
18 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
19 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
20 *provider*, the string is formatted in accordance with the current system culture.

21  
22 [Note: See System.Single.ToString.]  
23  
24  
25

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToString(System.Double) Method

```
4 [ILAsm]  
5 .method public hidebysig static string ToString(float64 value)  
6 [C#]  
7 public static string ToString(double value)
```

#### 8 Summary

9 Converts a System.Double to a System.String.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

11

#### 12 Return Value

13 Returns the value returned by *value*.ToString().

#### 14 Description

15 [Note: See System.Double.ToString.]  
16  
17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToString(System.Double, 4 System.IFormatProvider) Method

```
5 [ILAsm]  
6 .method public hidebysig static string ToString(float64 value, class  
7 System.IFormatProvider provider)  
  
8 [C#]  
9 public static string ToString(double value, IFormatProvider provider)
```

### 10 Summary

11 Converts a System.Double to a System.String.

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### 14 Return Value

15 Returns the value returned by *value*.ToString(*provider*).

### 16 Description

17 This method converts *value* using the information in the  
18 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
19 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
20 *provider*, the string is formatted in accordance with the current system culture.

21  
22 [Note: See System.Double.ToString.]  
23  
24  
25

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 Convert.ToString(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static string ToString(decimal value)  
6 [C#]  
7 public static string ToString(decimal value)
```

#### 8 Summary

9 Converts a System.Decimal to a System.String.

#### 10 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

11

#### 12 Return Value

13 Returns the value returned by *value*.ToString().

#### 14 Description

15 [Note: See System.Decimal.ToString.]  
16  
17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToString(System.Decimal, 4 System.IFormatProvider) Method

```
5 [ILAsm]  
6 .method public hidebysig static string ToString(decimal value, class  
7 System.IFormatProvider provider)  
8 [C#]  
9 public static string ToString(decimal value, IFormatProvider provider)
```

### 10 Summary

11 Converts a System.Decimal to a System.String.

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

### 14 Return Value

15 Returns the value returned by *value*.ToString(*provider*).

### 16 Description

17 This method converts *value* using the information in the  
18 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
19 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
20 *provider*, the string is formatted in accordance with the current system culture.

21  
22 [Note: See System.Decimal.ToString.]  
23  
24  
25

# 1 Convert.ToString(System.DateTime) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(valuetype System.DateTime  
4 value)  
5 [C#]  
6 public static string ToString(DateTime value)
```

## 7 Summary

8 Converts a System.DateTime value to a System.String.

## 9 Parameters

Parameter	Description
<i>value</i>	The System.DateTime value to be converted.

10

## 11 Return Value

12 Returns the value returned by *value*.ToString().

## 13 Description

14 [Note: See System.DateTime.ToString.]  
15  
16

17

# 1 Convert.ToString(System.DateTime, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static string ToString(valuetype System.DateTime  
5 value, class System.IFormatProvider provider)  
  
6 [C#]  
7 public static string ToString(DateTime value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.DateTime value to a System.String.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.DateTime value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.DateTimeFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 Returns the value returned by *value.ToString(provider)*.

## 14 Description

15 This method converts *value* using the information in the  
16 System.Globalization.DateTimeFormatInfo instance supplied by *provider*. If *provider*  
17 is null or if a System.Globalization.DateTimeFormatInfo cannot be obtained from  
18 *provider*, the string is formatted in accordance with the current system culture.

19

20 [Note: See System.DateTime.ToString.]

21

22

23

# 1 Convert.ToString(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(unsigned int32 value)  
4 [C#]  
5 public static string ToString(uint value)
```

## 6 Summary

7 Converts a System.UInt32 to a System.String.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString().

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToString(System.Int64).

17

18 [*Note:* See System.UInt32.ToString.]

19

20

21

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

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

## 8 Summary

9 Converts a System.Int32 to a System.String.

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 Returns the value returned by *value.ToString(provider)*.

## 14 Description

15 This method converts *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is formatted in accordance with the current system culture.

19

20 [Note: See System.Int32.ToString.]

21

22

23

# 1 Convert.ToString(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(int32 value)  
4 [C#]  
5 public static string ToString(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.String.

## 8 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

9

## 10 Return Value

11 Returns the value returned by *value*.ToString().

## 12 Description

13 [Note: See System.Int32.ToString.]  
14  
15

16

# Convert.ToString(System.UInt16, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(unsigned int16 value,
class System.IFormatProvider provider)

[C#]
public static string ToString(ushort value, IFormatProvider provider)
```

## Summary

Converts a System.UInt16 to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## Return Value

Returns the value returned by *value*.ToString(*provider*).

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToString(System.Int32, System.IFormatProvider).

This method converts *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is formatted in accordance with the current system culture.

[Note: See System.UInt16.ToString.]

# 1 Convert.ToString(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(unsigned int16 value)  
4 [C#]  
5 public static string ToString(ushort value)
```

## 6 Summary

7 Converts a System.UInt16 to a System.String.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString().

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToString(System.Int32).

17

18 [*Note:* See System.UInt16.ToString.]

19

20

21

# 1 Convert.ToString(System.Int16, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static string ToString(int16 value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static string ToString(short value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.Int16 to a System.String.

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString(*provider*).

## 14 Description

15 This method converts *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is formatted in accordance with the current system culture.

19

20 [Note: See System.Int16.ToString.]

21

22

23

# 1 Convert.ToString(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(int16 value)  
4 [C#]  
5 public static string ToString(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.String.

## 8 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

9

## 10 Return Value

11 Returns the value returned by *value*.ToString().

## 12 Description

13 [Note: See System.Int16.ToString.]  
14  
15

16

# 1 Convert.ToString(System.Byte, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static string ToString(unsigned int8 value, class  
5 System.IFormatProvider provider)  
  
6 [C#]  
7 public static string ToString(byte value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.Byte to a System.String.

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString(*provider*).

## 14 Description

15 This method converts *value* using the information in the  
16 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
17 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
18 *provider*, the string is formatted in accordance with the current system culture.

19

20 [Note: See System.Byte.ToString.]

21

22

23

# 1 Convert.ToString(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(unsigned int8 value)  
4 [C#]  
5 public static string ToString(byte value)
```

## 6 Summary

7 Converts a System.Byte to a System.String.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

9

## 10 Return Value

11 Returns the value returned by *value*.ToString().

## 12 Description

13 [Note: See System.Byte.ToString.]  
14  
15

16

# Convert.ToString(System.SByte, System.IFormatProvider) Method

```
[ILAsm]  
.method public hidebysig static string ToString(int8 value, class  
System.IFormatProvider provider)  
  
[C#]  
public static string ToString(sbyte value, IFormatProvider provider)
```

## Summary

Converts a System.SByte to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

## Return Value

Returns the value returned by *value*.ToString(*provider*).

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToString(System.Int16, System.IFormatProvider).

This method converts *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is formatted in accordance with the current system culture.

[Note: See System.SByte.ToString.]

# 1 Convert.ToString(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(int8 value)  
4 [C#]  
5 public static string ToString(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.String.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 Returns the value returned by *value*.ToString().

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToString(System.Int16).

17

18 [*Note:* See System.SByte.ToString.]

19

20

21

# 1 Convert.ToString(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(valuetype System.Char  
4 value)  
5 [C#]  
6 public static string ToString(char value)
```

## 7 Summary

8 Converts a System.Char to a System.String.

## 9 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted.

10

## 11 Return Value

12 *value* as a System.String.

13

# 1 Convert.ToString(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(bool value)  
4 [C#]  
5 public static string ToString(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.String.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

9

## 10 Return Value

11 Returns the value returned by *value*.ToString().

## 12 Description

13 [Note: See System.Boolean.ToString.]  
14  
15

16

# 1 Convert.ToString(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig static string ToString(string value)  
4 [C#]  
5 public static string ToString(string value)
```

## 6 Summary

7 Returns the specified string.

## 8 Parameters

Parameter	Description
<i>value</i>	A System.String.

9

## 10 Return Value

11 *value* is returned unchanged.

## 12 Description

13 [Note: This method is provided for completeness.]  
14  
15

16

# Convert.ToInt16(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(string value,
class System.IFormatProvider provider)

[C#]
public static ushort ToUInt16(string value, IFormatProvider provider)
```

## Summary

Converts a System.String to a System.UInt16.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	An object that implements the System.IFormatProvider interface and supplies a System.Globalization.NumberFormatInfo instance containing culture-specific formatting information.

## Return Value

*value* as a 16-bit unsigned integer.

## Description

This method parses *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is parsed using the formatting information of the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt32(System.String, System.IFormatProvider).

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.UInt16.MaxValue</code> or less than <code>System.UInt16.MinValue</code> .

1

2

# Convert.ToInt16(System.String) Method

```
[ILAsm]  
.method public hidebysig static unsigned int16 ToInt16(string value)  
  
[C#]  
public static ushort ToInt16(string value)
```

## Summary

Converts a System.String to a System.UInt16.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.

## Return Value

*value* as a 16-bit unsigned integer.

## Description

This method parses *value* using the information in a System.Globalization.NumberFormatInfo instance initialized for the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt32(System.String).

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.UInt16.MaxValue or less than

	System.UInt16.MinValue.
--	-------------------------

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 Convert.ToInt16(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static unsigned int16 ToUInt16(decimal value)  
  
6 [C#]  
7 public static ushort ToUInt16(decimal value)
```

### 8 Summary

9 Converts a System.Decimal to a System.UInt16.

### 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

### 12 Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### 14 Return Value

15 *value* as a 16-bit unsigned integer. *value* is rounded prior to conversion.

### 16 Description

17 Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to  
18 the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

19  
20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToInt32(System.Decimal).

### 22 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt16(System.Double) Method

```
[ILAsm]  
.method public hidebysig static unsigned int16 ToUInt16(float64 value)  
  
[C#]  
public static ushort ToUInt16(double value)
```

### Summary

Converts a System.Double to a System.UInt16.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

*value* as a 16-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt32(System.Double).

### Exceptions

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt16(System.Single) Method

```
[ILAsm]  
.method public hidebysig static unsigned int16 ToUInt16(float32 value)  
  
[C#]  
public static ushort ToUInt16(float value)
```

### Summary

Converts a `System.Single` to a `System.UInt16`.

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a 16-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt32(System.Single)`.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.UInt16.MaxValue</code> or less than <code>System.UInt16.MinValue</code> .

# 1 Convert.ToInt16(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(unsigned int64  
4 value)  
5 [C#]  
6 public static ushort ToUInt16(ulong value)
```

## 7 Summary

8 Converts a System.UInt64 to a System.UInt16.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a 16-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt32(System.Decimal).

## 18 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt16.MaxValue.

19

20

# 1 Convert.ToInt16(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToInt16(int64 value)  
4 [C#]  
5 public static ushort ToInt16(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.UInt16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue.

18

19

# 1 Convert.ToInt16(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(unsigned int32  
4 value)  
5 [C#]  
6 public static ushort ToUInt16(uint value)
```

## 7 Summary

8 Converts a System.UInt32 to a System.UInt16.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a 16-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt32(System.Int64).

## 18 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt16.MaxValue.

19

20

# 1 Convert.ToInt16(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(unsigned int16  
4 value)  
5 [C#]  
6 public static ushort ToUInt16(ushort value)
```

## 7 Summary

8 Converts a System.UInt16 to a System.UInt16.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* is returned unchanged.

## 15 Description

16 [Note: This method is provided for completeness.]

17

18

19

20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToInt32(System.Int32).

22

# 1 Convert.ToInt16(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToInt16(int32 value)  
4 [C#]  
5 public static ushort ToInt16(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.UInt16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

## 12 Return Value

13 value as a 16-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int32).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue.

18

19

# 1 Convert.ToInt16(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(int16 value)  
4 [C#]  
5 public static ushort ToUInt16(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.UInt16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt16.MinValue.

18

19

# 1 Convert.ToInt16(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(unsigned int8  
4 value)  
5 [C#]  
6 public static ushort ToUInt16(byte value)
```

## 7 Summary

8 Converts a System.Byte to a System.UInt16.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

12

## 13 Return Value

14 *value* as a 16-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt32(System.Byte).

18

# 1 Convert.ToInt16(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToUInt16(int8 value)  
4 [C#]  
5 public static ushort ToUInt16(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.UInt16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 16-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt16.MinValue.

18

19

# 1 Convert.ToInt16(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToInt16(valuetype  
4 System.Char value)  
  
5 [C#]  
6 public static ushort ToInt16(char value)
```

## 7 Summary

8 Converts a System.Char to a System.UInt16.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

12

## 13 Return Value

14 *value* as a 16-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt32(System.Char).

18

# 1 Convert.ToInt16(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int16 ToInt16(bool value)  
4 [C#]  
5 public static ushort ToInt16(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.UInt16.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

## 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt32(System.Boolean).

17

# 1 Convert.ToInt32(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToInt32(bool value)  
4 [C#]  
5 public static uint ToInt32(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.UInt32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

## 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Boolean).

17

# 1 Convert.ToInt32(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(valuetype  
4 System.Char value)  
  
5 [C#]  
6 public static uint ToUInt32(char value)
```

## 7 Summary

8 Converts a System.Char to a System.UInt32.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

12

## 13 Return Value

14 *value* as a 32-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt64(System.Char).

18

# 1 Convert.ToInt32(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(int8 value)  
4 [C#]  
5 public static uint ToUInt32(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.UInt32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt32.MinValue.

18

19

# 1 Convert.ToInt32(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(unsigned int8  
4 value)  
5 [C#]  
6 public static uint ToUInt32(byte value)
```

## 7 Summary

8 Converts a System.Byte to a System.UInt32.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

12

## 13 Return Value

14 *value* as a 32-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt64(System.Byte).

18

# 1 Convert.ToInt32(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(int16 value)  
4 [C#]  
5 public static uint ToUInt32(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.UInt32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt32.MinValue.

18

19

# 1 Convert.ToInt32(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(unsigned int16  
4 value)  
5 [C#]  
6 public static uint ToUInt32(ushort value)
```

## 7 Summary

8 Converts a System.UInt16 to a System.UInt32.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a 32-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToInt64(System.Int32).

18

# 1 Convert.ToInt32(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(int32 value)  
4 [C#]  
5 public static uint ToUInt32(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.UInt32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int32).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt32.MinValue.

18

19

# 1 Convert.ToInt32(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(unsigned int32  
4 value)  
5 [C#]  
6 public static uint ToUInt32(uint value)
```

## 7 Summary

8 Converts a System.UInt32 to a System.UInt32.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* is returned unchanged.

## 15 Description

16 [Note: This method is provided for completeness.]

17

18

19

20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToInt64(System.Int64).

22

# 1 Convert.ToInt32(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int32 ToUInt32(int64 value)  
4 [C#]  
5 public static uint ToUInt32(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.UInt32.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 32-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToInt64(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue.

18

19

# Convert.ToInt32(System.UInt64) Method

```
[ILAsm]  
.method public hidebysig static unsigned int32 ToUInt32(unsigned int64  
value)  
  
[C#]  
public static uint ToUInt32(ulong value)
```

## Summary

Converts a System.UInt64 to a System.UInt32.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

## Return Value

*value* as a 32-bit unsigned integer.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt64(System.Decimal).

## Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than System.UInt32.MaxValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt32(System.Single) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(float32 value)

[C#]
public static uint ToUInt32(float value)
```

### Summary

Converts a `System.Single` to a `System.UInt32`.

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a 32-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt64(System.Single)`.

### Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is greater than <code>System.UInt32.MaxValue</code> or less than <code>System.UInt32.MinValue</code> .

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt32(System.Double) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(float64 value)

[C#]
public static uint ToUInt32(double value)
```

### Summary

Converts a System.Double to a System.UInt32.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

*value* as a 32-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt64(System.Double).

### Exceptions

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt32(System.Decimal) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(decimal value)

[C#]
public static uint ToUInt32(decimal value)
```

### Summary

Converts a System.Decimal to a System.UInt32.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### Return Value

*value* as a 32-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt64(System.Decimal).

### Exceptions

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue.

# Convert.ToInt32(System.String) Method

```
[ILAsm]  
.method public hidebysig static unsigned int32 ToUInt32(string value)  
  
[C#]  
public static uint ToUInt32(string value)
```

## Summary

Converts a `System.String` to a `System.UInt32`.

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to be converted. The string is in the <code>System.Globalization.NumberStyles.Integer</code> style.

## Return Value

*value* as a 32-bit unsigned integer.

## Description

This method parses *value* using the information in a `System.Globalization.NumberFormatInfo` instance initialized for the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt64(System.String)`.

## Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>value</i> is a null reference.
<code>System.FormatException</code>	<i>value</i> cannot be converted to a numeric value.
<code>System.OverflowException</code>	The numeric value of <i>value</i> is greater than <code>System.UInt32.MaxValue</code> or less than

	System.UInt32.MinValue.
--	-------------------------

1

2

# Convert.ToInt32(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToInt32(string value,
class System.IFormatProvider provider)

[C#]
public static uint ToInt32(string value, IFormatProvider provider)
```

## Summary

Converts a System.String to a System.UInt32.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	An object that implements the System.IFormatProvider interface and supplies a System.Globalization.NumberFormatInfo instance containing culture-specific formatting information.

## Return Value

*value* as an 32-bit unsigned integer.

## Description

This method parses *value* using the information in the System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is null or if a System.Globalization.NumberFormatInfo cannot be obtained from *provider*, the string is parsed using the formatting information of the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToInt64(System.String, System.IFormatProvider).

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.UInt32.MaxValue</code> or less than <code>System.UInt32.MinValue</code> .

1

2

# 1 Convert.ToInt64(System.String, 2 System.IFormatProvider) Method

```
3 [ILAsm]  
4 .method public hidebysig static unsigned int64 ToUInt64(string value,  
5 class System.IFormatProvider provider)  
  
6 [C#]  
7 public static ulong ToUInt64(string value, IFormatProvider provider)
```

## 8 Summary

9 Converts a System.String to a System.UInt64.

## 10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

## 12 Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.Object that implements the System.IFormatProvider interface and supplies a System.Globalization.NumberFormatInfo instance containing culture-specific formatting information.

13

## 14 Return Value

15 *value* as a 64-bit unsigned integer.

## 16 Description

17 This method parses *value* using the information in the  
18 System.Globalization.NumberFormatInfo instance supplied by *provider*. If *provider* is  
19 null or if a System.Globalization.NumberFormatInfo cannot be obtained from  
20 *provider*, the string is parsed using the formatting information of the current system  
21 culture.

22

23 This member is not CLS-compliant. For a CLS-compliant alternative, use  
24 System.Convert.ToDecimal(System.Int64, System.IFormatProvider).

## 25 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is a null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than <code>System.UInt64.MaxValue</code> or less than <code>System.UInt64.MinValue</code> .

1

2

# Convert.ToInt64(System.String) Method

```
[ILAsm]  
.method public hidebysig static unsigned int64 ToUInt64(string value)  
  
[C#]  
public static ulong ToUInt64(string value)
```

## Summary

Converts a System.String to a System.UInt64.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.

## Return Value

*value* as a 64-bit unsigned integer.

## Description

This method parses *value* using the information in a System.Globalization.NumberFormatInfo instance initialized for the current system culture.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToDecimal(System.String).

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>value</i> is null reference.
<b>System.FormatException</b>	<i>value</i> cannot be converted to a numeric value.
<b>System.OverflowException</b>	The numeric value of <i>value</i> is greater than System.UInt64.MaxValue or less than

	System.UInt64.MinValue.
--	-------------------------

1

2

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt64(System.Decimal) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(decimal value)

[C#]
public static ulong ToUInt64(decimal value)
```

### Summary

Converts a System.Decimal to a System.UInt64.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

Parameter	Description
<i>value</i>	The System.Decimal value to be converted.

### Return Value

*value* as a 64-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToDecimal(System.Decimal).

### Exceptions

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt64(System.Double) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(float64 value)

[C#]
public static ulong ToUInt64(double value)
```

### Summary

Converts a System.Double to a System.UInt64.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

Parameter	Description
<i>value</i>	The System.Double value to be converted.

### Return Value

*value* as a 64-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use System.Convert.ToDecimal(System.Double).

### Exceptions

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## Convert.ToInt64(System.Single) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(float32 value)

[C#]
public static ulong ToUInt64(float value)
```

### Summary

Converts a `System.Single` to a `System.UInt64`.

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to be converted.

### Return Value

*value* as a 64-bit unsigned integer. *value* is rounded prior to conversion.

### Description

Prior to the conversion, if *value* is halfway between two whole numbers, it is rounded to the nearest even integer. For example, 4.5 is rounded to 4, and 5.5 is rounded to 6.

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToDecimal(System.Single)`.

### Exceptions

Exception	Condition
<code>System.OverflowException</code>	<i>value</i> is greater than <code>System.UInt64.MaxValue</code> or less than <code>System.UInt64.MinValue</code> .

# 1 Convert.ToInt64(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(unsigned int64  
4 value)  
5 [C#]  
6 public static ulong ToUInt64(ulong value)
```

## 7 Summary

8 Converts a System.UInt64 to a System.UInt64.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* is returned unchanged.

## 15 Description

16 [Note: This method is provided for completeness.]

17

18

19

20 This member is not CLS-compliant. For a CLS-compliant alternative, use  
21 System.Convert.ToDecimal(System.Decimal).

22

# 1 Convert.ToInt64(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(int64 value)  
4 [C#]  
5 public static ulong ToUInt64(long value)
```

## 6 Summary

7 Converts a System.Int64 to a System.UInt64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 64-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToDecimal(System.Int64).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt64.MinValue.

18

19

# 1 Convert.ToInt64(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(unsigned int32  
4 value)  
5 [C#]  
6 public static ulong ToUInt64(uint value)
```

## 7 Summary

8 Converts a System.UInt32 to a System.UInt64.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a 64-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Int64).

18

# 1 Convert.ToInt64(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(int32 value)  
4 [C#]  
5 public static ulong ToUInt64(int value)
```

## 6 Summary

7 Converts a System.Int32 to a System.UInt64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToDecimal(System.Int32).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt64.MinValue.

18

19

# 1 Convert.ToInt64(System.UInt16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(unsigned int16  
4 value)  
5 [C#]  
6 public static ulong ToUInt64(ushort value)
```

## 7 Summary

8 Converts a System.UInt16 to a System.UInt64.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

12

## 13 Return Value

14 *value* as a 64-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Int32).

18

# 1 Convert.ToInt64(System.Int16) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(int16 value)  
4 [C#]  
5 public static ulong ToUInt64(short value)
```

## 6 Summary

7 Converts a System.Int16 to a System.UInt64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToDecimal(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	<i>value</i> is less than System.UInt64.MinValue.

18

19

# 1 Convert.ToInt64(System.Byte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(unsigned int8  
4 value)  
5 [C#]  
6 public static ulong ToUInt64(byte value)
```

## 7 Summary

8 Converts a System.Byte to a System.UInt64.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Byte value to be converted.

12

## 13 Return Value

14 *value* as a 64-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Byte).

18

# 1 Convert.ToInt64(System.SByte) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(int8 value)  
4 [C#]  
5 public static ulong ToUInt64(sbyte value)
```

## 6 Summary

7 Converts a System.SByte to a System.UInt64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.SByte value to be converted.

11

## 12 Return Value

13 *value* as a 64-bit unsigned integer.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToDecimal(System.Int16).

## 17 Exceptions

Exception	Condition
<b>System.OverflowException</b>	value is less than System.UInt64.MinValue.

18

19

# 1 Convert.ToInt64(System.Char) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(valuetype  
4 System.Char value)  
5 [C#]  
6 public static ulong ToUInt64(char value)
```

## 7 Summary

8 Converts a System.Char to a System.UInt64.

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

Parameter	Description
<i>value</i>	The System.Char to be converted interpreted as an unsigned value.

12

## 13 Return Value

14 *value* as a 64-bit unsigned integer.

## 15 Description

16 This member is not CLS-compliant. For a CLS-compliant alternative, use  
17 System.Convert.ToDecimal(System.Char).

18

# 1 Convert.ToInt64(System.Boolean) Method

```
2 [ILAsm]  
3 .method public hidebysig static unsigned int64 ToUInt64(bool value)  
4 [C#]  
5 public static ulong ToUInt64(bool value)
```

## 6 Summary

7 Converts a System.Boolean to a System.UInt64.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to be converted.

11

## 12 Return Value

13 If *value* is true returns 1; if *value* is false returns 0.

## 14 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 System.Convert.ToDecimal(System.Boolean).

17