# System.Convert Class

```
[ILAsm]
.class public sealed Convert extends System.Object

[C#]
public sealed class Convert
```

### Assembly Info:

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

### Summary

Performs conversions between base data types.

### Inherits From: System.Object

**Library:** BCL

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

### Description

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

| **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 |

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

**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 |
|---|---|
| *value* | 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


(decimal) 1 as bool = True


(decimal) -2 as bool = True
```

**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** |
|---|---|
| *value* | 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

(double) 1 as bool = True

(double) -2 as bool = True
```

**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 |
|---|---|
| *value* | 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

(float) 1 as bool = True

(float) -2 as bool = True
```

# Convert.ToBoolean(System.String) Method

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

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

### Summary

Converts a `System.String` to a `System.Boolean`.

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.String` to be converted. |

### Return Value

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* is not equal to `System.Boolean.TrueString` or `System.Boolean.FalseString`. |

### Example

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

[C#]

```
using System;
class ConvertBoolTest {
    static public void Main() {
        string string0 = Boolean.TrueString;
        string string1 = Boolean.FalseString;
        string string2 = "foo"; //This is an invalid Boolean.
        bool bool0 = Convert.ToBoolean(string0);
        bool bool1 = Convert.ToBoolean(string1);
        Console.WriteLine("(string) {0} as bool = {1}",string0,bool0);
        Console.WriteLine("(string) {0} as bool = {1}",string1,bool1);
        bool bool2 = Convert.ToBoolean(string2); //Throws an exception.
        Console.WriteLine("(string) {0} as bool = {1}",string2,bool2);
    }
}
```

The output is

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

# 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 |
|---|---|
| *value* | 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);
    }
}
```

The output is

```
(ulong) 0 as bool = False


(ulong) 1 as bool = True
```

# 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** |
|---|---|
| *value* | 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

(long) -2 as bool = True
```

# 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 |
|---|---|
| *value* | 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);
    }
}
```

The output is

```
(uint) 0 as bool = False


(uint) 1 as bool = True
```

# 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 |
|---|---|
| *value* | 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

(int) -2 as bool = True
```

# 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 |
|---|---|
| *value* | 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);
    }
}
```

The output is

```
(ushort) 0 as bool = False


(ushort) 1 as bool = True
```

# 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 |
|---|---|
| *value* | 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

(short) -2 as bool = True
```

# Convert.ToBoolean(System.Byte) Method

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

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

### Summary

Converts a `System.Byte` to a `System.Boolean`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` 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() {
        byte byte0 = (byte) 0;
        byte byte1 = (Byte) 1;
        bool bool0 = Convert.ToBoolean(byte0);
        bool bool1 = Convert.ToBoolean(byte1);
        Console.WriteLine("(byte) {0} as bool = {1}",byte0,bool0);
        Console.WriteLine("(byte) {0} as bool = {1}",byte1,bool1);
    }
}
```

The output is

```
(byte) 0 as bool = False


(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 |
|---|---|
| *value* | 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);
        Console.WriteLine("(sbyte) {0} as bool = {1}",sbyte2,bool2);
    }
}
```

The output is

```
(sbyte) 0 as bool = False


(sbyte) 1 as bool = True


(sbyte) -2 as bool = True
```

# Convert.ToBoolean(System.Boolean) Method

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

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

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

# 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |

### Return Value

*value* as a `System.Byte`.

### Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

### Example

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

[C#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        string string0 = "+22";
        string string1 = "0";
        string string2 = "-1";
        byte byte0 = Convert.ToByte(string0);
        byte byte1 = Convert.ToByte(string1);
        Console.WriteLine("(string) {0} as byte = {1}",string0,byte0);
        Console.WriteLine("(string) {0} as byte = {1}",string1,byte1);
        byte byte2 = Convert.ToByte(string2);
        Console.WriteLine("(string) {0} as byte = {1}",string2,byte2);
    }
}
```

The output is

```
(string) +22 as byte = 22


(string) 0 as byte = 0


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


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


at System.Byte.Parse(String s)


at Convert.ToByte(String value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertString.cs:line 11
```

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

# Convert.ToByte(System.Decimal) Method

[ILAsm]

```
.method public hidebysig static unsigned int8 ToByte(decimal value)
```

[C#]

```
public static byte ToByte(decimal value)
```

## Summary

Converts a `System.Decimal` to a `System.Byte`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` 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. [*Note:* This process is known as banker's rounding.]

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`. |

## Example

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

[C#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        decimal decimal0 = 0.0m;
        decimal decimal1 = 1.5m;
        decimal decimal2 = 2.5m;
        decimal decimal3 = -1.0m;
        byte byte0 = Convert.ToByte(decimal0);
        byte byte1 = Convert.ToByte(decimal1);
        byte byte2 = Convert.ToByte(decimal2);
        Console.WriteLine("(decimal) {0} as byte =
{1}",decimal0,byte0);
        Console.WriteLine("(decimal) {0} as byte =
{1}",decimal1,byte1);
        Console.WriteLine("(decimal) {0} as byte =
{1}",decimal2,byte2);

        byte byte3 = Convert.ToByte(decimal3); //Throws an exception.
        Console.WriteLine("(decimal) {0} as byte =
{1}",decimal3,byte3);
    }
}
```

The output is

```
(decimal) 0 as byte = 0


(decimal) 1.5 as byte = 2


(decimal) 2.5 as byte = 2


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


at System.Decimal.ToByte(Decimal value)


at Convert.ToByte(Decimal value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertDecimal.cs:line 15
```

**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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`, or *value* 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#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        double double0 = 0.0;
        double double1 = 1.5;
        double double2 = 2.5;
        double double3 = -1.0;
        byte byte0 = Convert.ToByte(double0);
        byte byte1 = Convert.ToByte(double1);
        byte byte2 = Convert.ToByte(double2);
        Console.WriteLine("(double) {0} as byte = {1}",double0,byte0);
        Console.WriteLine("(double) {0} as byte = {1}",double1,byte1);
        Console.WriteLine("(double) {0} as byte = {1}",double2,byte2);

        byte byte3 = Convert.ToByte(double3); //Throws an exception.
        Console.WriteLine("(double) {0} as byte = {1}",double3,byte3);
    }
}
```

The output is

```
(double) 0 as byte = 0


(double) 1.5 as byte = 2


(double) 2.5 as byte = 2


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


at Convert.ToByte(Int32 value)


at Convert.ToByte(Double value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertDouble.cs:line 15
```

**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 |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`, or *value* is equal to one of `System.Single.NaN`, `System.Single.PositiveInfinity`, or `System.Single.NegativeInfinity`. |

### Example

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

[C#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        float float0 = 0.0f;
        float float1 = 1.5f;
        float float2 = 2.5f;
        float float3 = -1.0f;
        byte byte0 = Convert.ToByte(float0);
        byte byte1 = Convert.ToByte(float1);
        byte byte2 = Convert.ToByte(float2);
        Console.WriteLine("(float) {0} as byte = {1}",float0,byte0);
        Console.WriteLine("(float) {0} as byte = {1}",float1,byte1);
        Console.WriteLine("(float) {0} as byte = {1}",float2,byte2);

        byte byte3 = Convert.ToByte(float3); //Throws an exception.
        Console.WriteLine("(float) {0} as byte = {1}",float3,byte3);
    }
}
```

The output is

```
(float) 0 as byte = 0


(float) 1.5 as byte = 2


(float) 2.5 as byte = 2


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


at Convert.ToByte(Int32 value)


at Convert.ToByte(Single value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertFloat.cs:line 15
```

# Convert.ToByte(System.UInt64) Method

[ILAsm]

```
.method public hidebysig static unsigned int8 ToByte(unsigned int64
value)
```

[C#]

```
public static byte ToByte(ulong value)
```

### Summary

Converts a `System.UInt64` to a `System.Byte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-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.Decimal`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue`. |

### Example

The following example demonstrates converting `System.UInt64` values to `System.Byte` values.

[C#]

```
using System;
```

```
class ConvertByteTest {
    static public void Main() {
        ulong ulong0 = 0;
        ulong ulong1 = 32000;
        byte byte0 = Convert.ToByte(ulong0);
        Console.WriteLine("(ulong) {0} as byte = {1}",ulong0,byte0);
        byte byte1 = Convert.ToByte(ulong1); //Throws an exception.
        Console.WriteLine("(ulong) {0} as byte = {1}",ulong1,byte1);
    }
}
```

The output is

```
(ulong) 0 as byte = 0


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


at Convert.ToByte(UInt64 value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertUInt64.cs:line 8
```

# Convert.ToByte(System.Int64) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(int64 value)

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

### Summary

Converts a `System.Int64` to a `System.Byte`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Byte`.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`. |

# Convert.ToByte(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(unsigned int32
value)

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

## Summary

Converts a `System.UInt32` to a `System.Byte`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 32-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.Int64`).

## Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue`. |

# Convert.ToByte(System.Int32) Method

```
[ILAsm]
.method public hidebysig static unsigned int8 ToByte(int32 value)

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

## Summary

Converts a `System.Int32` to a `System.Byte`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Byte`.

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`. |

# 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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue`. |

### Example

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

[C#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        ushort ushort0 = 0;
        ushort ushort1 = 32000;
        byte byte0 = Convert.ToByte(ushort0);
        Console.WriteLine("(ushort) {0} as byte = {1}",ushort0,byte0);
        byte byte1 = Convert.ToByte(ushort1); //Throws an exception.
        Console.WriteLine("(ushort) {0} as byte = {1}",ushort1,byte1);
    }
}
```

The output is

```
(ushort) 0 as byte = 0


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


at Convert.ToByte(UInt16 value)


at ConvertByteTest.Main() in
C:\ECMAExamples\ConvertToByte\ConvertUInt16.cs:line 8
```

# Convert.ToByte(System.Int16) Method

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

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

### Summary

Converts a `System.Int16` to a `System.Byte`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Byte`.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Byte.MaxValue` or less than `System.Byte.MinValue`. |

# Convert.ToByte(System.SByte) Method

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

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

## Summary

Converts a `System.SByte` to a `System.Byte`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 8-bit signed integer 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.Int16`).

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.Byte.MinValue`. |

# Convert.ToByte(System.Boolean) Method

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

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

## Summary

Converts a `System.Boolean` to a `System.Byte`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

## Return Value

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

# Convert.ToByte(System.Byte) Method

[ILAsm]

```
.method public hidebysig static unsigned int8 ToByte(unsigned int8 value)
```

[C#]

```
public static byte ToByte(byte value)
```

## Summary

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

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

# Convert.ToByte(System.Char) Method

[ILAsm]

```
.method public hidebysig static unsigned int8 ToByte(valuetype
System.Char value)
```

[C#]

```
public static byte ToByte(char value)
```

## Summary

Converts a `System.Char` to a `System.Byte`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The Unicode character to be converted interpreted as an unsigned value. |

## Return Value

*value* as a `System.Byte`.

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | The numeric value of *value* is greater than `System.Byte.MaxValue`. |

## Example

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

[C#]

```
using System;
class ConvertByteTest {
    static public void Main() {
        char char0 = '0';
        char char1 = '1';
        char char2 = 'a';
        byte byte0 = Convert.ToByte(char0);
        byte byte1 = Convert.ToByte(char1);
        byte byte2 = Convert.ToByte(char2);
        Console.WriteLine("(char) {0} as byte = {1}",char0,byte0);
        Console.WriteLine("(char) {0} as byte = {1}",char1,byte1);
        Console.WriteLine("(char) {0} as byte = {1}",char2,byte2);
    }
}
```

The output is

```
(char) 0 as byte = 48


(char) 1 as byte = 49


(char) a as byte = 97
```

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

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

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

### Summary

Converts a `System.String` to a `System.Byte`.

### Parameters

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

### Return Value

*value* as a `System.Byte`.

### 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** | *value* is a null reference. |
| **System.FormatException** | *value* 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.` |

# Convert.ToChar(System.String) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char ToChar(string
value)

[C#]
public static char ToChar(string value)
```

## Summary

Converts a `System.String` to a `System.Char`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.String` to be converted. The `System.String` is required to contain a single character. |

## Return Value

*value* as a `System.Char`.

## Exceptions

| Exception | Condition |
|---|---|
| **System.FormatException** | *value* does not contain exactly one character. |

# Convert.ToChar(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char
ToChar(unsigned int64 value)

[C#]
public static char ToChar(ulong value)
```

### Summary

Converts a `System.UInt64` to a `System.Char`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Char`.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Char.MaxValue`. |

# Convert.ToChar(System.Int64) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char ToChar(int64
value)

[C#]
public static char ToChar(long value)
```

### Summary

Converts a `System.Int64` to a `System.Char`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Char`.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Char.MaxValue` or less than `System.Char.MinValue`. |

# Convert.ToChar(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char
ToChar(unsigned int32 value)

[C#]
public static char ToChar(uint value)
```

### Summary

Converts a `System.UInt32` to a `System.Char`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Char`.

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Char.MaxValue`. |

# Convert.ToChar(System.Int32) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char ToChar(int32
value)

[C#]
public static char ToChar(int value)
```

### Summary

Converts a `System.Int32` to a `System.Char`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Char`.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Char.MaxValue` or less than `System.Char.MinValue`. |

# Convert.ToChar(System.UInt16) Method

[ILAsm]
```
.method public hidebysig static valuetype System.Char
ToChar(unsigned int16 value)
```

[C#]
```
public static char ToChar(ushort value)
```

### Summary

Converts a `System.UInt16` to a `System.Char`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Char`.

### Description

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

# Convert.ToChar(System.Int16) Method

[ILAsm]
```
.method public hidebysig static valuetype System.Char ToChar(int16
value)
```

[C#]
```
public static char ToChar(short value)
```

## Summary

Converts a `System.Int16` to a `System.Char`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Char`.

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.Char.MinValue`. |

# Convert.ToChar(System.Byte) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char
ToChar(unsigned int8 value)

[C#]
public static char ToChar(byte value)
```

## Summary

Converts a `System.Byte` to a `System.Char`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

*value* as a `System.Char`.

# Convert.ToChar(System.SByte) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char ToChar(int8
value)

[C#]
public static char ToChar(sbyte value)
```

### Summary

Converts a `System.SByte` to a `System.Char`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a `System.Char`.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | value is less than `System.Char.MinValue`. |

# Convert.ToChar(System.Char) Method

```
[ILAsm]
.method public hidebysig static valuetype System.Char
ToChar(valuetype System.Char value)

[C#]
public static char ToChar(char value)
```

## Summary

Converts a `System.Char` to a `System.Char`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The Unicode character to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

# Convert.ToDateTime(System.DateTime) Method

[ILAsm]

```
.method public hidebysig static valuetype System.DateTime
ToDateTime(valuetype System.DateTime value)
```

[C#]

```
public static DateTime ToDateTime(DateTime value)
```

### Summary

Converts a `System.DateTime` to a `System.DateTime`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.DateTime` to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

# Convert.ToDateTime(System.String) Method

```
[ILAsm]
.method public hidebysig static valuetype System.DateTime
ToDateTime(string value)

[C#]
public static DateTime ToDateTime(string value)
```

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.String` to be converted. The string is in a form allowed by the `System.DateTime.Parse`(`System.String`) method. |

### Return Value

*value* as a `System.DateTime`.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a `System.DateTime`. |

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

```
[ILAsm]
.method public hidebysig static valuetype System.DateTime
ToDateTime(string value, class System.IFormatProvider provider)

[C#]
public static DateTime ToDateTime(string value, IFormatProvider
provider)
```

### Summary

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

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.String` to be converted. |
| *provider* | A `System.IFormatProvider` that supplies a `System.Globalization.DateTimeFormatInfo` containing culture-specific formatting information. |

### Return Value

*value* as a `System.DateTime`.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.ArgumentException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a `System.DateTime`. |

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

# Convert.ToDecimal(System.Boolean) Method

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

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

## Summary

Converts a `System.Boolean` to a `System.Decimal`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

## Return Value

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

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

# Convert.ToDecimal(System.Decimal) Method

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

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

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

**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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Number` style. |
| *provider* | 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 |
|---|---|
| **System.ArgumentException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than |

| | |
|---|---|
| | `System.Decimal.MaxValue` or less than `System.Decimal.MinValue`. |

**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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.ArgumentException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.Decimal.MaxValue` or less than `System.Decimal.MinValue`. |

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

# Convert.ToDecimal(System.Double) Method

```
[ILAsm]
.method public hidebysig static decimal ToDecimal(float64 value)

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

## Summary

Converts a `System.Double` to a `System.Decimal`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` value to be converted. |

## Return Value

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | The numeric value of *value* is greater than `System.Decimal.MaxValue` or less than `System.Decimal.MinValue`. |

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

# Convert.ToDecimal(System.Single) Method

```
[ILAsm]
.method public hidebysig static decimal ToDecimal(float32 value)

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

### Summary

Converts a `System.Single` to a `System.Decimal`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` value to be converted. |

### Return Value

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | The numeric value of *value* is greater than `System.Decimal.MaxValue` or less than `System.Decimal.MinValue`. |

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

# Convert.ToDecimal(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static decimal ToDecimal(unsigned int64
value)

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

### Summary

Converts a `System.UInt64` to a `System.Decimal`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Decimal`.

### Description

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

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

# Convert.ToDecimal(System.Int64) Method

[ILAsm]
```
.method public hidebysig static decimal ToDecimal(int64 value)
```

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

## Summary

Converts a `System.Int64` to a `System.Decimal`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Decimal`.

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

# Convert.ToDecimal(System.UInt32) Method

[ILAsm]
```
.method public hidebysig static decimal ToDecimal(unsigned int32 value)
```

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

### Summary

Converts a `System.UInt32` to a `System.Decimal`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Decimal`.

### Description

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

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

# Convert.ToDecimal(System.Int32) Method

[ILAsm]
```
.method public hidebysig static decimal ToDecimal(int32 value)
```

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

## Summary

Converts a `System.Int32` to a `System.Decimal`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Decimal`.

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

# Convert.ToDecimal(System.UInt16) Method

[ILAsm]
```
.method public hidebysig static decimal ToDecimal(unsigned int16 value)
```

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

### Summary

Converts a `System.UInt16` to a `System.Decimal`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Decimal`.

### Description

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

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

# Convert.ToDecimal(System.Int16) Method

```
[ILAsm]
.method public hidebysig static decimal ToDecimal(int16 value)

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

## Summary

Converts a `System.Int16` to a `System.Decimal`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Decimal`.

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

# Convert.ToDecimal(System.Byte) Method

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

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

### Summary

Converts a `System.Byte` to a `System.Decimal`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

### Return Value

*value* as a `System.Decimal`.

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

# Convert.ToDecimal(System.SByte) Method

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

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

### Summary

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

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a `System.Decimal`.

### Description

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

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

# Convert.ToDouble(System.Double) Method

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

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

### Summary

Converts a `System.Double` to a `System.Double`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

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

# Convert.ToDouble(System.Single) Method

[ILAsm]
```
.method public hidebysig static float64 ToDouble(float32 value)
```

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

## Summary

Converts a `System.Single` to a `System.Double`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` value to be converted. |

## Return Value

*value* as a `System.Double`.

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

# Convert.ToDouble(System.UInt64) Method

[ILAsm]

```
.method public hidebysig static float64 ToDouble(unsigned int64
value)
```

[C#]

```
public static double ToDouble(ulong value)
```

### Summary

Converts a `System.UInt64` to a `System.Double`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Double`.

### Description

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

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

# Convert.ToDouble(System.Int64) Method

[ILAsm]

```
.method public hidebysig static float64 ToDouble(int64 value)
```

[C#]

```
public static double ToDouble(long value)
```

### Summary

Converts a `System.Int64` to a `System.Double`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Double`.

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

# Convert.ToDouble(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static float64 ToDouble(unsigned int32
value)

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

### Summary

Converts a `System.UInt32` to a `System.Double`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Double`.

### Description

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

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

# Convert.ToDouble(System.Int32) Method

```
[ILAsm]
.method public hidebysig static float64 ToDouble(int32 value)

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

### Summary

Converts a `System.Int32` to a `System.Double`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

*value* as a `System.Double`.

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

# Convert.ToDouble(System.UInt16) Method

[ILAsm]
```
.method public hidebysig static float64 ToDouble(unsigned int16 value)
```

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

## Summary

Converts a `System.UInt16` to a `System.Double`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

## Return Value

*value* as a `System.Double`.

## Description

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

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

# Convert.ToDouble(System.Int16) Method

```
[ILAsm]
.method public hidebysig static float64 ToDouble(int16 value)

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

## Summary

Converts a `System.Int16` to a `System.Double`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Double`.

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

# Convert.ToDouble(System.Byte) Method

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

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

### Summary

Converts a `System.Byte` to a `System.Double`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

### Return Value

*value* as a `System.Double`.

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

# Convert.ToDouble(System.SByte) Method

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

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

### Summary

Converts a `System.SByte` to a `System.Double`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a `System.Double`.

### Description

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

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

# Convert.ToDouble(System.Boolean) Method

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

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

## Summary

Converts a `System.Boolean` to a `System.Double`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

## Return Value

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

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

# Convert.ToDouble(System.Decimal) Method

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

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

### Summary

Converts a `System.Decimal` to a `System.Double`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` value to be converted. |

### Return Value

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

**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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.Double.MaxValue` or less than `System.Double.MinValue`. |

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

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

```
[ILAsm]
.method public hidebysig static float64 ToDouble(string value, class
System.IFormatProvider provider)

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

### Summary

Converts a `System.String` to a `System.Double`.

### Parameters

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

### Return Value

*value* as a `System.Double`.

### 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.ArgumentException** | *value* is a null reference. |

| | |
|---|---|
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.Double.MaxValue` or less than `System.Double.MinValue`. |

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

# Convert.ToInt16(System.Decimal) Method

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

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

### Summary

Converts a `System.Decimal` to a `System.Int16`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue` or less than `System.Int16.MinValue`. |

# Convert.ToInt16(System.String) Method

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

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

## Summary

Converts a `System.String` to a `System.Int16`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |

## Return Value

*value* as a 16-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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

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

```
[ILAsm]
.method public hidebysig static int16 ToInt16(string value, class
System.IFormatProvider provider)

[C#]
public static short ToInt16(string value, IFormatProvider provider)
```

### Summary

Converts a `System.String` to a `System.Int16`.

### Parameters

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

### Return Value

*value* as a 16-bit signed 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.

### Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

**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 int16 ToInt16(float64 value)

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

## Summary

Converts a `System.Double` to a `System.Int16`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` 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 |
|---|---|
| **System.OverflowException** | *value* 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 |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue` or less than `System.Int16.MinValue`. |

# Convert.ToInt16(System.UInt64) Method

[ILAsm]
```
.method public hidebysig static int16 ToInt16(unsigned int64 value)
```

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

### Summary

Converts a `System.UInt64` to a `System.Int16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a 16-bit signed integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue`. |

# Convert.ToInt16(System.Int64) Method

[ILAsm]
```
.method public hidebysig static int16 ToInt16(int64 value)
```

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

## Summary

Converts a `System.Int64` to a `System.Int16`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

## Return Value

*value* as a 16-bit signed integer.

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | value is greater than `System.Int16.MaxValue` or less than `System.Int16.MinValue`. |

# Convert.ToInt16(System.Int16) Method

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

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

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

# Convert.ToInt16(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static int16 ToInt16(unsigned int32 value)

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

### Summary

Converts a `System.UInt32` to a `System.Int16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a 16-bit signed integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue`. |

# Convert.ToInt16(System.Int32) Method

```
[ILAsm]
.method public hidebysig static int16 ToInt16(int32 value)

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

### Summary

Converts a `System.Int32` to a `System.Int16`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

*value* as a 16-bit signed integer.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue` or less than `System.Int16.MinValue`. |

# Convert.ToInt16(System.UInt16) Method

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

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

## Summary

Converts a `System.UInt16` to a `System.Int16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

## Return Value

*value* as a 16-bit signed integer.

## Description

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int16.MaxValue`. |

# Convert.ToInt16(System.Byte) Method

[ILAsm]
```
.method public hidebysig static int16 ToInt16(unsigned int8 value)
```

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

### Summary

Converts a `System.Byte` to a `System.Int16`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

### Return Value

*value* as a 16-bit signed integer.

# Convert.ToInt16(System.SByte) Method

[ILAsm]
```
.method public hidebysig static int16 ToInt16(int8 value)
```

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

### Summary

Converts a `System.SByte` to a `System.Int16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a 16-bit signed integer.

### Description

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

# Convert.ToInt16(System.Char) Method

```
[ILAsm]
.method public hidebysig static int16 ToInt16(valuetype System.Char
value)

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

### Summary

Converts a `System.Char` to a `System.Int16`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

### Return Value

*value* as a 16-bit signed integer.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | The numeric value of *value* is greater than `System.Int16.MaxValue`. |

# Convert.ToInt16(System.Boolean) Method

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

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

### Summary

Converts a `System.Boolean` to a `System.Int16`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

# Convert.ToInt32(System.Boolean) Method

[ILAsm]
```
.method public hidebysig static int32 ToInt32(bool value)
```

[C#]
```
public static int ToInt32(bool value)
```

### Summary

Converts a `System.Boolean` to a `System.Int32`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

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

```
[ILAsm]
.method public hidebysig static int32 ToInt32(string value, class
System.IFormatProvider provider)

[C#]
public static int ToInt32(string value, IFormatProvider provider)
```

## Summary

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

## Parameters

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

## Return Value

*value* as a 32-bit signed 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.

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

# 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

**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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* 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.Double) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(float64 value)

[C#]
public static int ToInt32(double value)
```

### Summary

Converts a `System.Double` to a `System.Int32`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` 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 |
|---|---|
| **System.OverflowException** | *value* 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 |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int32.MaxValue` or less than `System.Int32.MinValue`. |

# Convert.ToInt32(System.UInt64) Method

[ILAsm]
```
.method public hidebysig static int32 ToInt32(unsigned int64 value)
```

[C#]
```
public static int ToInt32(ulong value)
```

### Summary

Converts a `System.UInt64` to a `System.Int32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int32.MaxValue`. |

# Convert.ToInt32(System.Int64) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(int64 value)

[C#]
public static int ToInt32(long value)
```

### Summary

Converts a `System.Int64` to a `System.Int32`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int32.MaxValue` or less than `System.Int32.MinValue`. |

# Convert.ToInt32(System.Int32) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(int32 value)

[C#]
public static int ToInt32(int value)
```

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

# Convert.ToInt32(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(unsigned int32 value)

[C#]
public static int ToInt32(uint value)
```

### Summary

Converts a `System.UInt32` to a `System.Int32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int32.MaxValue`. |

# Convert.ToInt32(System.UInt16) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(unsigned int16 value)

[C#]
public static int ToInt32(ushort value)
```

### Summary

Converts a `System.UInt16` to a `System.Int32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

### Description

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

# Convert.ToInt32(System.Int16) Method

[ILAsm]

```
.method public hidebysig static int32 ToInt32(int16 value)
```

[C#]

```
public static int ToInt32(short value)
```

### Summary

Converts a `System.Int16` to a `System.Int32`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

# Convert.ToInt32(System.Byte) Method

```
[ILAsm]
.method public hidebysig static int32 ToInt32(unsigned int8 value)

[C#]
public static int ToInt32(byte value)
```

## Summary

Converts a `System.Byte` to a `System.Int32`.

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

*value* as a 32-bit signed integer.

# Convert.ToInt32(System.SByte) Method

[ILAsm]
```
.method public hidebysig static int32 ToInt32(int8 value)
```

[C#]
```
public static int ToInt32(sbyte value)
```

### Summary

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

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a 32-bit signed integer.

### Description

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

# Convert.ToInt32(System.Char) Method

[ILAsm]

```
.method public hidebysig static int32 ToInt32(valuetype System.Char
value)
```

[C#]

```
public static int ToInt32(char value)
```

### Summary

Converts a `System.Char` to a `System.Int32`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

### Return Value

*value* as a 32-bit signed integer.

# Convert.ToInt64(System.Char) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(valuetype System.Char
value)
```

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

### Summary

Converts a `System.Char` to a `System.Int64`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

### Return Value

*value* as a 64-bit signed integer.

# Convert.ToInt64(System.Boolean) Method

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

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

### Summary

Converts a `System.Boolean` to a `System.Int64`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

# Convert.ToInt64(System.SByte) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(int8 value)
```

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

### Summary

Converts a `System.SByte` to a `System.Int64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a 64-bit signed integer.

### Description

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

# Convert.ToInt64(System.Byte) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(unsigned int8 value)
```

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

## Summary

Converts a `System.Byte` to a `System.Int64`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

*value* as a 64-bit signed integer.

# Convert.ToInt64(System.Int16) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(int16 value)
```

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

## Summary

Converts a `System.Int16` to a `System.Int64`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

## Return Value

*value* as a 64-bit signed integer.

# Convert.ToInt64(System.UInt16) Method

[ILAsm]

```
.method public hidebysig static int64 ToInt64(unsigned int16 value)
```

[C#]

```
public static long ToInt64(ushort value)
```

## Summary

Converts a `System.UInt16` to a `System.Int64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

## Return Value

*value* as a 64-bit signed integer.

## Description

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

# Convert.ToInt64(System.Int32) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(int32 value)
```

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

## Summary

Converts a `System.Int32` to a `System.Int64`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

## Return Value

*value* as a 64-bit signed integer.

# Convert.ToInt64(System.UInt32) Method

[ILAsm]
```
.method public hidebysig static int64 ToInt64(unsigned int32 value)
```

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

### Summary

Converts a `System.UInt32` to a `System.Int64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a 64-bit signed integer.

### Description

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

# Convert.ToInt64(System.UInt64) Method

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

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

## Summary

Converts a `System.UInt64` to a `System.Int64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

## Return Value

*value* as a 64-bit signed integer.

## Description

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Int64.MaxValue`. |

# Convert.ToInt64(System.Int64) Method

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

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

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

**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 int64 ToInt64(float32 value)

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

### Summary

Converts a `System.Single` to a `System.Int64`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* 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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* 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.Decimal) Method

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

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

### Summary

Converts a `System.Decimal` to a `System.Int64`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` 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 |
|---|---|
| **System.OverflowException** | *value* 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

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

```
[ILAsm]
.method public hidebysig static int64 ToInt64(string value, class
System.IFormatProvider provider)

[C#]
public static long ToInt64(string value, IFormatProvider provider)
```

### Summary

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

### Parameters

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

### Return Value

*value* as a 64-bit signed 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.

### Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

# Convert.ToSByte(System.Boolean) Method

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

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

### Summary

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

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

### Description

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

# Convert.ToSByte(System.SByte) Method

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

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

### Summary

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

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* is returned unchanged.

### Description

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

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

# Convert.ToSByte(System.Char) Method

```
[ILAsm]
.method public hidebysig static int8 ToSByte(valuetype System.Char
value)

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

### Summary

Converts a `System.Char` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

### Return Value

*value* as a `System.SByte`.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | The numeric value of *value* is greater than `System.SByte.MaxValue`. |

# Convert.ToSByte(System.Byte) Method

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

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

### Summary

Converts a `System.Byte` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` 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.Byte`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue`. |

# Convert.ToSByte(System.Int16) Method

[ILAsm]
```
.method public hidebysig static int8 ToSByte(int16 value)
```

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

### Summary

Converts a `System.Int16` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-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.Int16`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue` or less than `System.SByte.MinValue`. |

# 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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue`. |

# Convert.ToSByte(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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue` or less than `System.SByte.MinValue`. |

# Convert.ToSByte(System.UInt32) Method

[ILAsm]
```
.method public hidebysig static int8 ToSByte(unsigned int32 value)
```

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

## Summary

Converts a `System.UInt32` to a `System.SByte`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 32-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.Int64`).

## Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue`. |

# Convert.ToSByte(System.Int64) Method

[ILAsm]
```
.method public hidebysig static int8 ToSByte(int64 value)
```

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

### Summary

Converts a `System.Int64` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-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.Int64`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue` or less than `System.SByte.MinValue`. |

# Convert.ToSByte(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static int8 ToSByte(unsigned int64 value)

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

### Summary

Converts a `System.UInt64` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-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.Decimal`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue`. |

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

# Convert.ToSByte(System.Single) Method

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

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

### Summary

Converts a `System.Single` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` 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.Single`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* 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.Double) Method

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

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

### Summary

Converts a `System.Double` to a `System.SByte`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` 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.Double`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* 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 |
|---|---|
| *value* | The `System.Decimal` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.SByte.MaxValue` or less than `System.SByte.MinValue`. |

# 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* 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`. |

# 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |
| *provider* | 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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Float` \| `System.Globalization.NumberStyles.AllowThousands` style. |
| *provider* | 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** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than |

| | |
|---|---|
| | `System.Single.MaxValue` or less than `System.Single.MinValue`. |

**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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.Single.MaxValue` or less than `System.Single.MinValue`. |

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

# Convert.ToSingle(System.Decimal) Method

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

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

## Summary

Converts a `System.Decimal` to a `System.Single`.

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Decimal` value to be converted. |

## Return Value

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

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

# Convert.ToSingle(System.Double) Method

```
[ILAsm]
.method public hidebysig static float32 ToSingle(float64 value)

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

## Summary

Converts a `System.Double` to a `System.Single`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` value to be converted. |

## Return Value

*value* as a `System.Single`.

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.Single.MaxValue` or less than `System.Single.MinValue`. |

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

# Convert.ToSingle(System.Single) Method

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

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

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` value to be converted. |

## Return Value

*value* as a `System.Single`.

## Description

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

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

# Convert.ToSingle(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static float32 ToSingle(unsigned int64
value)

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

### Summary

Converts a `System.UInt64` to a `System.Single`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Single`.

### Description

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

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

# Convert.ToSingle(System.Int64) Method

```
[ILAsm]
.method public hidebysig static float32 ToSingle(int64 value)

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

## Summary

Converts a `System.Int64` to a `System.Single`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Single`.

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

# Convert.ToSingle(System.UInt32) Method

[ILAsm]
```
.method public hidebysig static float32 ToSingle(unsigned int32
value)
```

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

### Summary

Converts a `System.UInt32` to a `System.Single`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Single`.

### Description

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

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

# Convert.ToSingle(System.Int32) Method

[ILAsm]
```
.method public hidebysig static float32 ToSingle(int32 value)
```

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

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Single`.

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

# Convert.ToSingle(System.UInt16) Method

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

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

### Summary

Converts a `System.UInt16` to a `System.Single`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

### Return Value

*value* as a `System.Single`.

### Description

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

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

# Convert.ToSingle(System.Int16) Method

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

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

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

## Return Value

*value* as a `System.Single`.

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

# Convert.ToSingle(System.Byte) Method

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

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

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

### Return Value

*value* as a `System.Single`.

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

# Convert.ToSingle(System.SByte) Method

[ILAsm]

```
.method public hidebysig static float32 ToSingle(int8 value)
```

[C#]

```
public static float ToSingle(sbyte value)
```

### Summary

Converts a `System.SByte` to a `System.Single`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a `System.Single`.

### Description

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

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

# Convert.ToSingle(System.Boolean) Method

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

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

### Summary

Converts a `System.Boolean` to a `System.Single`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

# 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 |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |
| *provider* | 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`.]

# Convert.ToString(System.Int64) Method

[ILAsm]
```
.method public hidebysig static string ToString(int64 value)
```

[C#]
```
public static string ToString(long value)
```

## Summary

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

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

[*Note:* See `System.Int64.ToString`.]

# Convert.ToString(System.Int64, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(int64 value, class
System.IFormatProvider provider)

[C#]
public static string ToString(long value, IFormatProvider provider)
```

### Summary

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

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |
| *provider* | 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 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.Int64.ToString`.]

# Convert.ToString(System.UInt64) Method

[ILAsm]
```
.method public hidebysig static string ToString(unsigned int64
value)
```

[C#]
```
public static string ToString(ulong value)
```

## Summary

Converts a `System.UInt64` to a `System.String`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

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

[*Note:* See `System.UInt64.ToString`.]

# 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 |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |
| *provider* | 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`.]

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

# Convert.ToString(System.Single) Method

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

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

## Summary

Converts a `System.Single` to a `System.String`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

[*Note:* See `System.Single.ToString`.]

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

# Convert.ToString(System.Single, System.IFormatProvider) Method

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

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

## Summary

Converts a `System.Single` to a `System.String`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Single` value to be converted. |
| *provider* | 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 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.Single.ToString`.]

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

# Convert.ToString(System.Double) Method

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

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

## Summary

Converts a `System.Double` to a `System.String`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

[*Note:* See `System.Double.ToString`.]

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

# Convert.ToString(System.Double, System.IFormatProvider) Method

[ILAsm]
```
.method public hidebysig static string ToString(float64 value, class System.IFormatProvider provider)
```

[C#]
```
public static string ToString(double value, IFormatProvider provider)
```

## Summary

Converts a `System.Double` to a `System.String`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Double` value to be converted. |
| *provider* | 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 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.Double.ToString`.]

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

# Convert.ToString(System.Decimal) Method

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

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

### Summary

Converts a `System.Decimal` to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` value to be converted. |

### Return Value

Returns the value returned by *value*.ToString().

### Description

[*Note:* See `System.Decimal.ToString`.]

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

# Convert.ToString(System.Decimal, System.IFormatProvider) Method

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

[C#]
```
public static string ToString(decimal value, IFormatProvider provider)
```

### Summary

Converts a `System.Decimal` to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` value to be converted. |
| *provider* | 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 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.Decimal.ToString`.]

# Convert.ToString(System.DateTime) Method

```
[ILAsm]
.method public hidebysig static string ToString(valuetype
System.DateTime value)

[C#]
public static string ToString(DateTime value)
```

### Summary

Converts a `System.DateTime` value to a `System.String`.

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.DateTime` value to be converted. |

### Return Value

Returns the value returned by *value*.ToString().

### Description

[*Note:* See `System.DateTime.ToString`.]

# Convert.ToString(System.DateTime, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(valuetype
System.DateTime value, class System.IFormatProvider provider)

[C#]
public static string ToString(DateTime value, IFormatProvider
provider)
```

### Summary

Converts a `System.DateTime` value to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.DateTime` value to be converted. |
| *provider* | A `System.IFormatProvider` that supplies a `System.Globalization.DateTimeFormatInfo` containing culture-specific formatting information. |

### Return Value

Returns the value returned by *value*.ToString(*provider*).

### Description

This method converts *value* using the information in the `System.Globalization.DateTimeFormatInfo` instance supplied by *provider*. If *provider* is `null` or if a `System.Globalization.DateTimeFormatInfo` cannot be obtained from *provider*, the string is formatted in accordance with the current system culture.

[*Note:* See `System.DateTime.ToString`.]

# Convert.ToString(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static string ToString(unsigned int32
value)

[C#]
public static string ToString(uint value)
```

## Summary

Converts a `System.UInt32` to a `System.String`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

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

[*Note:* See `System.UInt32.ToString`.]

# Convert.ToString(System.Int32, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(int32 value, class
System.IFormatProvider provider)

[C#]
public static string ToString(int value, IFormatProvider provider)
```

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |
| *provider* | 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 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.Int32.ToString`.]

# Convert.ToString(System.Int32) Method

[ILAsm]
```
.method public hidebysig static string ToString(int32 value)
```

[C#]
```
public static string ToString(int value)
```

## Summary

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

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

[*Note:* See `System.Int32.ToString`.]

# 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 |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |
| *provider* | 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`.]

# Convert.ToString(System.UInt16) Method

[ILAsm]
```
.method public hidebysig static string ToString(unsigned int16
value)
```

[C#]
```
public static string ToString(ushort value)
```

### Summary

Converts a `System.UInt16` to a `System.String`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

### Return Value

Returns the value returned by *value*.ToString().

### Description

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

[*Note:* See `System.UInt16.ToString`.]

# Convert.ToString(System.Int16, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static string ToString(int16 value, class
System.IFormatProvider provider)

[C#]
public static string ToString(short value, IFormatProvider provider)
```

## Summary

Converts a `System.Int16` to a `System.String`.

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |
| *provider* | 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 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.Int16.ToString`.]

# Convert.ToString(System.Int16) Method

[ILAsm]
```
.method public hidebysig static string ToString(int16 value)
```

[C#]
```
public static string ToString(short value)
```

### Summary

Converts a `System.Int16` to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

Returns the value returned by *value*.ToString().

### Description

[*Note:* See `System.Int16.ToString`.]

# Convert.ToString(System.Byte, System.IFormatProvider) Method

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

[C#]
```
public static string ToString(byte value, IFormatProvider provider)
```

## Summary

Converts a `System.Byte` to a `System.String`.

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |
| *provider* | 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 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.Byte.ToString`.]

# Convert.ToString(System.Byte) Method

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

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

## Summary

Converts a `System.Byte` to a `System.String`.

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

[*Note:* See `System.Byte.ToString`.]

# 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 |
|---|---|
| *value* | The `System.SByte` value to be converted. |
| *provider* | 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`.]

# Convert.ToString(System.SByte) Method

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

[C#]
```
public static string ToString(sbyte value)
```

## Summary

Converts a `System.SByte` to a `System.String`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

## Return Value

Returns the value returned by *value*.ToString().

## Description

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

[*Note:* See `System.SByte.ToString`.]

# Convert.ToString(System.Char) Method

```
[ILAsm]
.method public hidebysig static string ToString(valuetype
System.Char value)

[C#]
public static string ToString(char value)
```

### Summary

Converts a `System.Char` to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted. |

### Return Value

*value* as a `System.String`.

# Convert.ToString(System.Boolean) Method

```
[ILAsm]
.method public hidebysig static string ToString(bool value)

[C#]
public static string ToString(bool value)
```

### Summary

Converts a `System.Boolean` to a `System.String`.

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

Returns the value returned by *value*.ToString().

### Description

[*Note:* See `System.Boolean.ToString`.]

# Convert.ToString(System.String) Method

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

[C#]
public static string ToString(string value)
```

## Summary

Returns the specified string.

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | A `System.String`. |

## Return Value

*value* is returned unchanged.

## Description

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

# Convert.ToUInt16(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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |
| *provider* | 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.UInt16.MaxValue` or less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.String) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(string
value)

[C#]
public static ushort ToUInt16(string value)
```

## Summary

Converts a `System.String` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* 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.ToUInt16(System.Decimal) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(decimal
value)

[C#]
public static ushort ToUInt16(decimal value)
```

## Summary

Converts a `System.Decimal` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Decimal` 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.Decimal`).

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* 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.ToUInt16(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 |
|---|---|
| *value* | 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** | *value* 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.ToUInt16(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 |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt16.MaxValue` or less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(unsigned
int64 value)

[C#]
public static ushort ToUInt16(ulong value)
```

## Summary

Converts a `System.UInt64` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

## Return Value

*value* as a 16-bit unsigned integer.

## Description

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt16.MaxValue`. |

# Convert.ToUInt16(System.Int64) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(int64 value)
```

[C#]
```
public static ushort ToUInt16(long value)
```

### Summary

Converts a `System.Int64` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

### Return Value

*value* as a 16-bit unsigned integer.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt16.MaxValue` or less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(unsigned
int32 value)

[C#]
public static ushort ToUInt16(uint value)
```

### Summary

Converts a `System.UInt32` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

### Return Value

*value* as a 16-bit unsigned integer.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt16.MaxValue`. |

# Convert.ToUInt16(System.UInt16) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(unsigned
int16 value)
```

[C#]
```
public static ushort ToUInt16(ushort value)
```

## Summary

Converts a `System.UInt16` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

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

# Convert.ToUInt16(System.Int32) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(int32 value)
```

[C#]
```
public static ushort ToUInt16(int value)
```

### Summary

Converts a `System.Int32` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

value as a 16-bit unsigned integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt16.MaxValue` or less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.Int16) Method

```
[ILAsm]
.method public hidebysig static unsigned int16 ToUInt16(int16 value)

[C#]
public static ushort ToUInt16(short value)
```

### Summary

Converts a `System.Int16` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

*value* as a 16-bit unsigned integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.Byte) Method

[ILAsm]

```
.method public hidebysig static unsigned int16 ToUInt16(unsigned
int8 value)
```

[C#]

```
public static ushort ToUInt16(byte value)
```

### Summary

Converts a `System.Byte` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

### Return Value

*value* as a 16-bit unsigned integer.

### Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt32`(`System.Byte`).

# Convert.ToUInt16(System.SByte) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(int8 value)
```

[C#]
```
public static ushort ToUInt16(sbyte value)
```

## Summary

Converts a `System.SByte` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` value to be converted. |

## Return Value

*value* as a 16-bit unsigned integer.

## Description

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

## Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt16.MinValue`. |

# Convert.ToUInt16(System.Char) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(valuetype
System.Char value)
```

[C#]
```
public static ushort ToUInt16(char value)
```

## Summary

Converts a `System.Char` to a `System.UInt16`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

## Return Value

*value* as a 16-bit unsigned integer.

## Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt32`(`System.Char`).

# Convert.ToUInt16(System.Boolean) Method

[ILAsm]
```
.method public hidebysig static unsigned int16 ToUInt16(bool value)
```

[C#]
```
public static ushort ToUInt16(bool value)
```

### Summary

Converts a `System.Boolean` to a `System.UInt16`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

### Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt32`(`System.Boolean`).

# Convert.ToUInt32(System.Boolean) Method

[ILAsm]
```
.method public hidebysig static unsigned int32 ToUInt32(bool value)
```

[C#]
```
public static uint ToUInt32(bool value)
```

### Summary

Converts a `System.Boolean` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

### Description

This member is not CLS-compliant. For a CLS-compliant alternative, use `System.Convert.ToInt64`(`System.Boolean`).

# Convert.ToUInt32(System.Char) Method

[ILAsm]
```
.method public hidebysig static unsigned int32 ToUInt32(valuetype
System.Char value)
```

[C#]
```
public static uint ToUInt32(char value)
```

## Summary

Converts a `System.Char` to a `System.UInt32`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

## 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.Char`).

# Convert.ToUInt32(System.SByte) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(int8 value)

[C#]
public static uint ToUInt32(sbyte value)
```

### Summary

Converts a `System.SByte` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.SByte` 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.Int16`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(System.Byte) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(unsigned
int8 value)

[C#]
public static uint ToUInt32(byte value)
```

### Summary

Converts a `System.Byte` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Byte` 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.Byte`).

# Convert.ToUInt32(System.Int16) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(int16 value)

[C#]
public static uint ToUInt32(short value)
```

### Summary

Converts a `System.Int16` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 16-bit signed 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.Int16`).

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(System.UInt16) Method

[ILAsm]

```
.method public hidebysig static unsigned int32 ToUInt32(unsigned
int16 value)
```

[C#]

```
public static uint ToUInt32(ushort value)
```

### Summary

Converts a `System.UInt16` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 16-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.Int32`).

# Convert.ToUInt32(System.Int32) Method

[ILAsm]
```
.method public hidebysig static unsigned int32 ToUInt32(int32 value)
```

[C#]
```
public static uint ToUInt32(int value)
```

### Summary

Converts a `System.Int32` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed 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.Int32`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(unsigned
int32 value)

[C#]
public static uint ToUInt32(uint value)
```

## Summary

Converts a `System.UInt32` to a `System.UInt32`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

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

# Convert.ToUInt32(System.Int64) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(int64 value)

[C#]
public static uint ToUInt32(long value)
```

### Summary

Converts a `System.Int64` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 64-bit signed 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.Int64`).

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt32.MaxValue` or less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(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 |
|---|---|
| *value* | 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 |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt32.MaxValue`. |

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

# Convert.ToUInt32(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 |
|---|---|
| *value* | The `System.Single` 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 |
|---|---|
| **System.OverflowException** | *value* 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.ToUInt32(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** |
|---|---|
| *value* | 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** | *value* 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.ToUInt32(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 |
|---|---|
| *value* | 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** | *value* is greater than `System.UInt32.MaxValue` or less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(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 |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` 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 |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.UInt32.MaxValue` or less than `System.UInt32.MinValue`. |

# Convert.ToUInt32(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static unsigned int32 ToUInt32(string
value, class System.IFormatProvider provider)

[C#]
public static uint ToUInt32(string value, IFormatProvider provider)
```

### Summary

Converts a `System.String` to a `System.UInt32`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |
| *provider* | 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** |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.UInt32.MaxValue` or less than `System.UInt32.MinValue`. |

# Convert.ToUInt64(System.String, System.IFormatProvider) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(string
value, class System.IFormatProvider provider)

[C#]
public static ulong ToUInt64(string value, IFormatProvider provider)
```

### Summary

Converts a `System.String` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.String` to be converted. The string is in the `System.Globalization.NumberStyles.Integer` style. |
| *provider* | A `System.Object` that implements the `System.IFormatProvider` interface and supplies a `System.Globalization.NumberFormatInfo` instance containing culture-specific formatting information. |

### Return Value

*value* as a 64-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.ToDecimal(System.Int64, System.IFormatProvider )`.

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.ArgumentNullException** | *value* is a null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* is greater than `System.UInt64.MaxValue` or less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(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** |
|---|---|
| *value* | 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** |
|---|---|
| **System.ArgumentNullException** | *value* is null reference. |
| **System.FormatException** | *value* cannot be converted to a numeric value. |
| **System.OverflowException** | The numeric value of *value* 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.ToUInt64(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 |
|---|---|
| *value* | 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** | *value* 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.ToUInt64(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 |
|---|---|
| *value* | 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** | *value* 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.ToUInt64(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** |
|---|---|
| *value* | The `System.Single` 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** |
|---|---|
| **System.OverflowException** | *value* is greater than `System.UInt64.MaxValue` or less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(System.UInt64) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(unsigned
int64 value)

[C#]
public static ulong ToUInt64(ulong value)
```

## Summary

Converts a `System.UInt64` to a `System.UInt64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 64-bit unsigned integer value to be converted. |

## Return Value

*value* is returned unchanged.

## Description

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

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

# Convert.ToUInt64(System.Int64) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(int64 value)

[C#]
public static ulong ToUInt64(long value)
```

## Summary

Converts a `System.Int64` to a `System.UInt64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 64-bit signed integer value to be converted. |

## Return Value

*value* as a 64-bit unsigned integer.

## Description

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

## Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(System.UInt32) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(unsigned
int32 value)

[C#]
public static ulong ToUInt64(uint value)
```

## Summary

Converts a `System.UInt32` to a `System.UInt64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit unsigned integer value to be converted. |

## Return Value

*value* as a 64-bit unsigned integer.

## Description

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

# Convert.ToUInt64(System.Int32) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(int32 value)

[C#]
public static ulong ToUInt64(int value)
```

### Summary

Converts a `System.Int32` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The 32-bit signed integer value to be converted. |

### Return Value

*value* as a 64-bit unsigned integer.

### Description

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

### Exceptions

| Exception | Condition |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(System.UInt16) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(unsigned
int16 value)

[C#]
public static ulong ToUInt64(ushort value)
```

## Summary

Converts a `System.UInt16` to a `System.UInt64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The 16-bit unsigned integer value to be converted. |

## Return Value

*value* as a 64-bit unsigned integer.

## Description

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

# Convert.ToUInt64(System.Int16) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(int16 value)

[C#]
public static ulong ToUInt64(short value)
```

### Summary

Converts a `System.Int16` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The 16-bit signed integer value to be converted. |

### Return Value

*value* as a 64-bit unsigned integer.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | *value* is less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(System.Byte) Method

[ILAsm]
```
.method public hidebysig static unsigned int64 ToUInt64(unsigned
int8 value)
```

[C#]
```
public static ulong ToUInt64(byte value)
```

## Summary

Converts a `System.Byte` to a `System.UInt64`.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Byte` value to be converted. |

## Return Value

*value* as a 64-bit unsigned integer.

## Description

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

# Convert.ToUInt64(System.SByte) Method

[ILAsm]
```
.method public hidebysig static unsigned int64 ToUInt64(int8 value)
```

[C#]
```
public static ulong ToUInt64(sbyte value)
```

### Summary

Converts a `System.SByte` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.SByte` value to be converted. |

### Return Value

*value* as a 64-bit unsigned integer.

### Description

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

### Exceptions

| **Exception** | **Condition** |
|---|---|
| **System.OverflowException** | value is less than `System.UInt64.MinValue`. |

# Convert.ToUInt64(System.Char) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(valuetype
System.Char value)

[C#]
public static ulong ToUInt64(char value)
```

### Summary

Converts a `System.Char` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| **Parameter** | **Description** |
|---|---|
| *value* | The `System.Char` to be converted interpreted as an unsigned value. |

### Return Value

*value* as a 64-bit unsigned integer.

### Description

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

# Convert.ToUInt64(System.Boolean) Method

```
[ILAsm]
.method public hidebysig static unsigned int64 ToUInt64(bool value)

[C#]
public static ulong ToUInt64(bool value)
```

### Summary

Converts a `System.Boolean` to a `System.UInt64`.

### Type Attributes:

- CLSCompliantAttribute(false)

### Parameters

| Parameter | Description |
|---|---|
| *value* | The `System.Boolean` value to be converted. |

### Return Value

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

### Description

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