

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]
- Version: 1.0.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 may 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

1
2
3
4
5
6
7
8
9

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 may 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.

10

1 Convert.ToBoolean(System.Boolean)

2 Method

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

7 Summary

8 Converts a **System.Boolean** to a **System.Boolean**.

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* is returned unchanged.

16 Description

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

18

1 Convert.ToBoolean(System.SByte) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11 Parameter	12 Description
<i>value</i>	The 8-bit signed integer value to be converted.

14 Return Value

15
16 **true** if *value* is non-zero; **false** if *value* is zero.

17 Description

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

20 Example

21
22 The following example demonstrates converting **System.SByte** values
23 to **System.Boolean** values.

```
24 [C#]  
25  
26 using System;  
27 class ConvertBoolTest {  
28     static public void Main() {  
29         sbyte sbyte0 = (sbyte) 0;  
30         sbyte sbyte1 = (sbyte) 1;  
31         sbyte sbyte2 = (sbyte) -2;  
32         bool bool0 = Convert.ToBoolean(sbyte0);  
33         bool bool1 = Convert.ToBoolean(sbyte1);  
34         bool bool2 = Convert.ToBoolean(sbyte2);
```

```
1         Console.WriteLine("(sbyte) {0} as bool =
2 {1}",sbyte0,bool0);
3         Console.WriteLine("(sbyte) {0} as bool =
4 {1}",sbyte1,bool1);
5         Console.WriteLine("(sbyte) {0} as bool =
6 {1}",sbyte2,bool2);
7     }
8 }
```

9 The output is

```
10
11 (sbyte) 0 as bool = False
12
13
14 (sbyte) 1 as bool = True
15
16
17 (sbyte) -2 as bool = True
18
```

19

1 Convert.ToBoolean(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.Boolean**.

9 Parameters

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

13 Return Value

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

16 Example

18 The following example demonstrates converting **System.Byte** values
19 to **System.Boolean** values.

```
20 [C#]  
21  
22 using System;  
23 class ConvertBoolTest {  
24     static public void Main() {  
25         byte byte0 = (byte) 0;  
26         byte byte1 = (Byte) 1;  
27         bool bool0 = Convert.ToBoolean(byte0);  
28         bool bool1 = Convert.ToBoolean(byte1);  
29         Console.WriteLine("(byte) {0} as bool =  
30 {1}", byte0, bool0);  
31         Console.WriteLine("(byte) {0} as bool =  
32 {1}", byte1, bool1);  
33     }  
34 }
```

35 The output is

```
36 (byte) 0 as bool = False  
37
```

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

1 Convert.ToBoolean(System.Int16) Method

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

6 Summary

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

8 Parameters

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

12 Return Value

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

15 Example

17 The following example demonstrates converting **System.Byte** values
18 to **System.Boolean** values.

```
19 [C#]  
20  
21 using System;  
22 class ConvertBoolTest {  
23     static public void Main() {  
24         short short0 = 0;  
25         short short1 = 1;  
26         short short2 = -2;  
27         bool bool0 = Convert.ToBoolean(short0);  
28         bool bool1 = Convert.ToBoolean(short1);  
29         bool bool2 = Convert.ToBoolean(short2);  
30         Console.WriteLine("(short) {0} as bool =  
31 {1}", short0, bool0);  
32         Console.WriteLine("(short) {0} as bool =  
33 {1}", short1, bool1);  
34         Console.WriteLine("(short) {0} as bool =  
35 {1}", short2, bool2);  
36     }  
37 }
```

```
1      The output is
2
3      (short) 0 as bool = False
4
5
6      (short) 1 as bool = True
7
8
9      (short) -2 as bool = True
10
```

11

Convert.ToBoolean(System.UInt16) Method

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

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

Summary

Converts a **System.UInt16** to a **System.Boolean**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

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

Return Value

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

Description

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

Example

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

```
[C#]

using System;
class ConvertBoolTest {
    static public void Main() {
        ushort ushort0 = 0;
        ushort ushort1 = 1;
        bool bool0 = Convert.ToBoolean(ushort0);
        bool bool1 = Convert.ToBoolean(ushort1);
```

```
1         Console.WriteLine("(ushort) {0} as bool =  
2 {1}",ushort0,bool0);  
3         Console.WriteLine("(ushort) {0} as bool =  
4 {1}",ushort1,bool1);  
5     }  
6 }
```

7 The output is

```
8  
9 (ushort) 0 as bool = False  
10  
11  
12 (ushort) 1 as bool = True  
13
```

14

1 Convert.ToBoolean(System.Int32) Method

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

6 Summary

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

8 Parameters

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

12 Return Value

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

15 Example

17 The following example demonstrates converting **System.Int32** values
18 to **System.Boolean** values.

```
19 [C#]  
20  
21 using System;  
22 class ConvertBoolTest {  
23     static public void Main() {  
24         int int0 = 0;  
25         int int1 = 1;  
26         int int2 = -2;  
27         bool bool0 = Convert.ToBoolean(int0);  
28         bool bool1 = Convert.ToBoolean(int1);  
29         bool bool2 = Convert.ToBoolean(int2);  
30         Console.WriteLine("(int) {0} as bool =  
31 {1}",int0,bool0);  
32         Console.WriteLine("(int) {0} as bool =  
33 {1}",int1,bool1);  
34         Console.WriteLine("(int) {0} as bool =  
35 {1}",int2,bool2);  
36     }  
37 }
```

```
1      The output is
2
3      (int) 0 as bool = False
4
5
6      (int) 1 as bool = True
7
8
9      (int) -2 as bool = True
10
```

11

Convert.ToBoolean(System.UInt32) Method

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

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

Summary

Converts a **System.UInt32** to a **System.Boolean**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

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

Return Value

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

Description

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

Example

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

```
[C#]

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

```
1         Console.WriteLine("(uint) {0} as bool =
2 {1}",uint0,bool0);
3         Console.WriteLine("(uint) {0} as bool =
4 {1}",uint1,bool1);
5     }
6 }
```

7 The output is

```
8
9 (uint) 0 as bool = False
10
11
12 (uint) 1 as bool = True
13
```

14

1 Convert.ToBoolean(System.Int64) Method

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

6 Summary

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

8 Parameters

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

12 Return Value

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

15 Example

17 The following example demonstrates converting **System.Int64** values
18 to **System.Boolean** values.

```
19 [C#]  
20  
21 using System;  
22 class ConvertBoolTest {  
23     static public void Main() {  
24         long long0 = 0;  
25         long long1 = 1;  
26         long long2 = -2;  
27         bool bool0 = Convert.ToBoolean(long0);  
28         bool bool1 = Convert.ToBoolean(long1);  
29         bool bool2 = Convert.ToBoolean(long2);  
30         Console.WriteLine("(long) {0} as bool =  
31 {1}", long0, bool0);  
32         Console.WriteLine("(long) {0} as bool =  
33 {1}", long1, bool1);  
34         Console.WriteLine("(long) {0} as bool =  
35 {1}", long2, bool2);  
36     }  
37 }
```

```
1      The output is
2
3      (long) 0 as bool = False
4
5
6      (long) 1 as bool = True
7
8
9      (long) -2 as bool = True
10
```

11

1 Convert.ToBoolean(System.UInt64)

2 Method

```
3 [ILASM]
4 .method public hidebysig static bool ToBoolean(unsigned
5 int64 value)
6
7 [C#]
8 public static bool ToBoolean(ulong value)
```

8 Summary

9 Converts a **System.UInt64** to a **System.Boolean**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19 Description

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

22 Example

23

24 The following example demonstrates converting **System.UInt64**
25 values to **System.Boolean** values.

26
27

```
28 [C#]
29 using System;
30 class ConvertBoolTest {
31     static public void Main() {
32         ulong ulong0 = 0;
33         ulong ulong1 = 1;
34         bool bool0 = Convert.ToBoolean(ulong0);
35         bool bool1 = Convert.ToBoolean(ulong1);
```

```
1         Console.WriteLine("(ulong) {0} as bool =  
2 {1}",ulong0,bool0);  
3         Console.WriteLine("(ulong) {0} as bool =  
4 {1}",ulong1,bool1);  
5     }  
6 }
```

```
7     The output is  
8  
9     (ulong) 0 as bool = False  
10  
11  
12     (ulong) 1 as bool = True  
13
```

14

1 Convert.ToBoolean(System.String)

2 Method

```
3 [ILASM]
4 .method public hidebysig static bool ToBoolean(string
5 value)
6
7 [C#]
8 public static bool ToBoolean(string value)
```

8 Summary

9 Converts a **System.String** to a **System.Boolean**.

10 Parameters

11
12

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

13
14
15

14 Return Value

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

18 Exceptions

19
20

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

21
22
23

22 Example

24 The following example demonstrates converting **System.String** values
25 to **System.Boolean** values.

26
27

```
28 [C#]
29 using System;
30 class ConvertBoolTest {
31     static public void Main() {
32         string string0 = Boolean.TrueString;
33         string string1 = Boolean.FalseString;
```

```
1         string string2 = "foo"; //This is an invalid
2 Boolean.
3         bool bool0 = Convert.ToBoolean(string0);
4         bool bool1 = Convert.ToBoolean(string1);
5         Console.WriteLine("(string) {0} as bool =
6 {1}",string0,bool0);
7         Console.WriteLine("(string) {0} as bool =
8 {1}",string1,bool1);
9         bool bool2 = Convert.ToBoolean(string2); //Throws
10 an exception.
11         Console.WriteLine("(string) {0} as bool =
12 {1}",string2,bool2);
13     }
14 }
```

15 The output is

```
16
17 (string) True as bool = True
18 (string) False as bool = False
19 Unhandled Exception: System.FormatException: String was not
20 recognized as a valid Boolean.
21     at System.Boolean.Parse(String value)
22     at Convert.ToBoolean(String value)
23     at ConvertBoolTest.Main() in
24 C:\ECMAExamples\ConvertString.cs:line 12
25
```

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

3 Convert.ToBoolean(System.Single) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static bool ToBoolean(float32  
7 value)  
8 [C#]  
9 public static bool ToBoolean(float value)
```

10 Summary

11 Converts a **System.Single** to a **System.Boolean**.

12 Parameters

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

16 Return Value

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

19 Example

21 The following example demonstrates converting **System.Single**
22 values to **System.Boolean** values.

```
23 [C#]  
24  
25 using System;  
26 class ConvertBoolTest {  
27     static public void Main() {  
28         float float0 = 0.0f;  
29         float float1 = 1.0f;  
30         float float2 = -2.0f;  
31         bool bool0 = Convert.ToBoolean(float0);  
32         bool bool1 = Convert.ToBoolean(float1);  
33         bool bool2 = Convert.ToBoolean(float2);  
34         Console.WriteLine("(float) {0} as bool =  
35 {1}",float0,bool0);  
36         Console.WriteLine("(float) {0} as bool =  
37 {1}",float1,bool1);
```

```
1         Console.WriteLine("(float) {0} as bool =  
2 {1}",float2,bool2);  
3     }  
4 }
```

```
5     The output is  
6  
7     (float) 0 as bool = False  
8  
9  
10    (float) 1 as bool = True  
11  
12  
13    (float) -2 as bool = True  
14
```

15

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

3 Convert.ToBoolean(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static bool ToBoolean(float64  
7 value)  
8 [C#]  
9 public static bool ToBoolean(double value)
```

10 Summary

11 Converts a **System.Double** to a **System.Boolean**.

12 Parameters

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

16 Return Value

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

19 Example

21 The following example demonstrates converting **System.Double**
22 values to **System.Boolean** values.

```
23 [C#]  
24  
25 using System;  
26 class ConvertBoolTest {  
27     static public void Main() {  
28         double double0 = 0.0;  
29         double double1 = 1.0;  
30         double double2 = -2.0;  
31         bool bool0 = Convert.ToBoolean(double0);  
32         bool bool1 = Convert.ToBoolean(double1);  
33         bool bool2 = Convert.ToBoolean(double2);  
34         Console.WriteLine("(double) {0} as bool =  
35 {1}",double0,bool0);  
36         Console.WriteLine("(double) {0} as bool =  
37 {1}",double1,bool1);
```

```
1         Console.WriteLine("(double) {0} as bool =  
2 {1}",double2,bool2);  
3     }  
4 }
```

5 The output is

```
6  
7 (double) 0 as bool = False
```

```
8  
9  
10 (double) 1 as bool = True
```

```
11  
12  
13 (double) -2 as bool = True
```

```
14
```

15

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

3 Convert.ToBoolean(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static bool ToBoolean(decimal  
7 value)  
8 [C#]  
9 public static bool ToBoolean(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.Boolean**.

12 Parameters

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

16 Return Value

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

19 Example

21 The following example demonstrates converting **System.Decimal**
22 values to **System.Boolean** values.

```
23 [C#]  
24  
25 using System;  
26 class ConvertBoolTest {  
27     static public void Main() {  
28         decimal decimal0 = 0m;  
29         decimal decimal1 = 1m;  
30         decimal decimal2 = -2m;  
31         bool bool0 = Convert.ToBoolean(decimal0);  
32         bool bool1 = Convert.ToBoolean(decimal1);  
33         bool bool2 = Convert.ToBoolean(decimal2);  
34         Console.WriteLine("(decimal) {0} as bool =  
35 {1}", decimal0, bool0);  
36         Console.WriteLine("(decimal) {0} as bool =  
37 {1}", decimal1, bool1);
```

```
1         Console.WriteLine("(decimal) {0} as bool =  
2 {1}", decimal2, bool2);  
3     }  
4 }
```

```
5     The output is  
6  
7     (decimal) 0 as bool = False  
8  
9  
10    (decimal) 1 as bool = True  
11  
12  
13    (decimal) -2 as bool = True  
14
```

15

1 Convert.ToByte(System.Boolean) Method

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

7 Summary

8 Converts a **System.Boolean** to a **System.Byte**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToByte(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.Byte**.

9 Parameters

10
11

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

12

13 Return Value

14

15 *value* is returned unchanged.

16 Description

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

18

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
<i>value</i>	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 <i>value</i> 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);
```

```
1         Console.WriteLine("(char) {0} as byte =
2 {1}",char0,byte0);
3         Console.WriteLine("(char) {0} as byte =
4 {1}",char1,byte1);
5         Console.WriteLine("(char) {0} as byte =
6 {1}",char2,byte2);
7     }
8 }
```

9

The output is

10

(char) 0 as byte = 48

11

12

13

(char) 1 as byte = 49

14

15

16

(char) a as byte = 97

17

18

19

1 Convert.ToByte(System.SByte) Method

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

7 Summary

8 Converts a **System.SByte** to a **System.Byte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

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

14 Return Value

15 *value* as a **System.Byte**.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToByte(System.Int16)**.

21 Exceptions

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

24
25
26

1 Convert.ToByte(System.Int16) Method

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

7 Summary

8 Converts a **System.Int16** to a **System.Byte**.

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* as a **System.Byte**.

16 Exceptions

17
18

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

19
20
21

Convert.ToByte(System.UInt16) Method

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

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

Summary

Converts a **System.UInt16** to a **System.Byte**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

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

Return Value

value as a **System.Byte**.

Description

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

Exceptions

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

Example

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

```
[C#]
```

```
1      using System;
2      class ConvertByteTest {
3          static public void Main() {
4              ushort ushort0 = 0;
5              ushort ushort1 = 32000;
6              byte byte0 = Convert.ToByte(ushort0);
7              Console.WriteLine("(ushort) {0} as byte =
8 {1}",ushort0,byte0);
9              byte byte1 = Convert.ToByte(ushort1); //Throws an
10 exception.
11              Console.WriteLine("(ushort) {0} as byte =
12 {1}",ushort1,byte1);
13          }
14      }
```

15 The output is

```
16
17      (ushort) 0 as byte = 0
18
```

```
19
20      Exception occurred: System.OverflowException: Value was
21      either too large or too small for an unsigned byte.
22
```

```
23
24      at Convert.ToByte(UInt16 value)
25
```

```
26
27      at ConvertByteTest.Main() in
28      C:\ECMAExamples\ConvertToByte\ConvertUInt16.cs:line 8
29
```

1 Convert.ToByte(System.Int32) Method

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

7 Summary

8 Converts a **System.Int32** to a **System.Byte**.

9 Parameters

10
11

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

12

13 Return Value

14

15 *value* as a **System.Byte**.

16 Exceptions

17
18

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

19

20

21

1 Convert.ToByte(System.UInt32) Method

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

7 Summary

8 Converts a **System.UInt32** to a **System.Byte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a **System.Byte**.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToByte(System.Int64)**.

21 Exceptions

22
23

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

24
25
26

1 Convert.ToByte(System.Int64) Method

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

7 Summary

8 Converts a **System.Int64** to a **System.Byte**.

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* as a **System.Byte**.

16 Exceptions

17
18

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

19
20
21

1 Convert.ToByte(System.UInt64) Method

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

7 Summary

8 Converts a **System.UInt64** to a **System.Byte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.Byte**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.Byte.MaxValue .

22 Example

23 The following example demonstrates converting **System.UInt64**
24 values to **System.Byte** values.

```
25 [C#]
```

```
1      using System;
2      class ConvertByteTest {
3          static public void Main() {
4              ulong ulong0 = 0;
5              ulong ulong1 = 32000;
6              byte byte0 = Convert.ToByte(ulong0);
7              Console.WriteLine("(ulong) {0} as byte =
8 {1}",ulong0,byte0);
9              byte byte1 = Convert.ToByte(ulong1); //Throws an
10 exception.
11              Console.WriteLine("(ulong) {0} as byte =
12 {1}",ulong1,byte1);
13          }
14      }
```

15 The output is

```
16
17      (ulong) 0 as byte = 0
18
```

```
19
20      Exception occurred: System.OverflowException: Value was
21      either too large or too small for an unsigned byte.
22
```

```
23
24      at Convert.ToByte(UInt64 value)
25
```

```
26
27      at ConvertByteTest.Main() in
28      C:\ECMAExamples\ConvertToByte\ConvertUInt64.cs:line 8
29
```

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

Convert.ToByte(System.Single) Method

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

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

Summary

Converts a **System.Single** to a **System.Byte**.

Parameters

Parameter	Description
<i>value</i>	The 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	<i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue , or <i>value</i> is equal to one of System.Single.NaN , System.Single.PositiveInfinity , or System.Single.NegativeInfinity .

Example

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

```

1
2      [C#]

3      using System;
4      class ConvertByteTest {
5          static public void Main() {
6              float float0 = 0.0f;
7              float float1 = 1.5f;
8              float float2 = 2.5f;
9              float float3 = -1.0f;
10             byte byte0 = Convert.ToByte(float0);
11             byte byte1 = Convert.ToByte(float1);
12             byte byte2 = Convert.ToByte(float2);
13             Console.WriteLine("(float) {0} as byte =
14 {1}",float0,byte0);
15             Console.WriteLine("(float) {0} as byte =
16 {1}",float1,byte1);
17             Console.WriteLine("(float) {0} as byte =
18 {1}",float2,byte2);
19
20             byte byte3 = Convert.ToByte(float3); //Throws an
21 exception.
22             Console.WriteLine("(float) {0} as byte =
23 {1}",float3,byte3);
24         }
25     }

```

26 The output is

```

27
28 (float) 0 as byte = 0
29
30
31 (float) 1.5 as byte = 2
32
33
34 (float) 2.5 as byte = 2
35

```

```

36
37 Exception occurred: System.OverflowException: Value was
38 either too large or too small for an unsigned byte.

```

```
1
2
3   at Convert.ToByte(Int32 value)
4
5
6   at Convert.ToByte(Single value)
7
8
9   at ConvertByteTest.Main() in
10  C:\ECMAExamples\ConvertToByte\ConvertFloat.cs:line 15
11
12
```

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

3 Convert.ToByte(System.Double) Method

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

9 Summary

10 Converts a **System.Double** to a **System.Byte**.

11 Parameters

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

15 Return Value

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

18 Description

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

23 Exceptions

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

27 Example

1 The following example demonstrates converting **System.Double**
2 values to **System.Byte** values.

```
3  
4 [C#]  
  
5 using System;  
6 class ConvertByteTest {  
7     static public void Main() {  
8         double double0 = 0.0;  
9         double double1 = 1.5;  
10        double double2 = 2.5;  
11        double double3 = -1.0;  
12        byte byte0 = Convert.ToByte(double0);  
13        byte byte1 = Convert.ToByte(double1);  
14        byte byte2 = Convert.ToByte(double2);  
15        Console.WriteLine("(double) {0} as byte =  
16 {1}",double0,byte0);  
17        Console.WriteLine("(double) {0} as byte =  
18 {1}",double1,byte1);  
19        Console.WriteLine("(double) {0} as byte =  
20 {1}",double2,byte2);  
21  
22        byte byte3 = Convert.ToByte(double3); //Throws an  
23 exception.  
24        Console.WriteLine("(double) {0} as byte =  
25 {1}",double3,byte3);  
26    }  
27 }
```

28 The output is

29
30 (double) 0 as byte = 0

31
32
33 (double) 1.5 as byte = 2

34
35
36 (double) 2.5 as byte = 2

37
38
39 Exception occurred: System.OverflowException: Value was

```
1      either too large or too small for an unsigned byte.
2
3
4      at Convert.ToByte(Int32 value)
5
6
7      at Convert.ToByte(Double value)
8
9
10     at ConvertByteTest.Main() in
11     C:\ECMAExamples\ConvertToByte\ConvertDouble.cs:line 15
12
13
```

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

3 Convert.ToByte(System.Decimal) Method

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

9 Summary

10 Converts a **System.Decimal** to a **System.Byte**.

11 Parameters

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

15 Return Value

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

18 Description

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

23 Exceptions

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

27 Example

29 The following example demonstrates converting **System.Decimal**
30 values to **System.Byte** values.

```
31 [C#]  
32
```

```
1      using System;
2      class ConvertByteTest {
3          static public void Main() {
4              decimal decimal0 = 0.0m;
5              decimal decimal1 = 1.5m;
6              decimal decimal2 = 2.5m;
7              decimal decimal3 = -1.0m;
8              byte byte0 = Convert.ToByte(decimal0);
9              byte byte1 = Convert.ToByte(decimal1);
10             byte byte2 = Convert.ToByte(decimal2);
11             Console.WriteLine("(decimal) {0} as byte =
12 {1}",decimal0,byte0);
13             Console.WriteLine("(decimal) {0} as byte =
14 {1}",decimal1,byte1);
15             Console.WriteLine("(decimal) {0} as byte =
16 {1}",decimal2,byte2);
17
18             byte byte3 = Convert.ToByte(decimal3); //Throws an
19 exception.
20             Console.WriteLine("(decimal) {0} as byte =
21 {1}",decimal3,byte3);
22         }
23     }
```

24 The output is

```
25
26 (decimal) 0 as byte = 0
27
28
29 (decimal) 1.5 as byte = 2
30
31
32 (decimal) 2.5 as byte = 2
33
```

```
34
35 Exception occurred: System.OverflowException: Value was
36 either too large or too small for an unsigned byte.
37
38
```

```
1      at System.Decimal.ToByte(Decimal value)
2
3
4      at Convert.ToByte(Decimal value)
5
6
7      at ConvertByteTest.Main() in
8      C:\ECMAExamples\ConvertToByte\ConvertDecimal.cs:line 15
9
10
```

1 Convert.ToByte(System.String) Method

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

7 Summary

8 Converts a **System.String** representation of a number to a
9 **System.Byte**.

10 Parameters

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

14 Return Value

16 *value* as a **System.Byte**.

17 Exceptions

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

21 Example

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

```
25 [C#]  
26  
27 using System;  
28 class ConvertByteTest {
```

```
1         static public void Main() {
2             string string0 = "+22";
3             string string1 = "0";
4             string string2 = "-1";
5             byte byte0 = Convert.ToByte(string0);
6             byte byte1 = Convert.ToByte(string1);
7             Console.WriteLine("(string) {0} as byte =
8 {1}",string0,byte0);
9             Console.WriteLine("(string) {0} as byte =
10 {1}",string1,byte1);
11             byte byte2 = Convert.ToByte(string2);
12             Console.WriteLine("(string) {0} as byte =
13 {1}",string2,byte2);
14         }
15     }
```

16 The output is

```
17
18 (string) +22 as byte = 22
19
```

```
20
21 (string) 0 as byte = 0
22
```

```
23
24 Exception occurred: System.OverflowException: Value was
25 either too large or too small for an unsigned byte.
```

```
26
27
28 at System.Byte.Parse(String s, NumberStyles style,
29 IFormatProvider provider)
```

```
30
31
32 at System.Byte.Parse(String s)
33
```

```
1
2   at Convert.ToByte(String value)
3
4
5   at ConvertByteTest.Main() in
6   C:\ECMAExamples\ConvertToByte\ConvertString.cs:line 11
7
8
```

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

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

9 Summary

10 Converts a **System.String** to a **System.Byte**.

11 Parameters

12
13

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

14
15
16

Return Value

17 *value* as a **System.Byte**.

18 Description

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

25 Exceptions

26
27

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

1
2
3

System.OverflowException	The numeric value of <i>value</i> is greater than System.Byte.MaxValue or less than System.Byte.MinValue .
---------------------------------	--

1 Convert.ToChar(System.Char) Method

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

7 Summary

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

9 Parameters

10
11

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

12
13
14

13 Return Value

15 *value* is returned unchanged.

16 Description

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

18

1 Convert.ToChar(System.SByte) Method

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

7 Summary

8 Converts a **System.SByte** to a **System.Char**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The System.SByte value to be converted.

14 Return Value

15 *value* as a **System.Char**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	value is less than System.Char.MinValue .

1 Convert.ToChar(System.Byte) Method

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

7 Summary

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

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* as a **System.Char**.

16

1 Convert.ToInt16(System.Int16) Method

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

7 Summary

8 Converts a **System.Int16** to a **System.Char**.

9 Parameters

10
11

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

12

13 Return Value

14

15 *value* as a **System.Char**.

16 Exceptions

17
18

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

19
20
21

1 Convert.ToChar(System.UInt16) Method

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

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a **System.Char**.

18 Description

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

21

1 Convert.ToInt32(Method)

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

7 Summary

8 Converts a **System.Int32** to a **System.Char**.

9 Parameters

10
11

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

12

13 Return Value

14

15 *value* as a **System.Char**.

16 Exceptions

17

18

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

19

20

21

1 Convert.ToChar(System.UInt32) Method

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

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.Char**.

18 Exceptions

19 Exception	20 Condition
System.OverflowException	<i>value</i> is greater than System.Char.MaxValue .

21
22
23

1 Convert.ToInt64(System.Int64) Method

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

7 Summary

8 Converts a **System.Int64** to a **System.Char**.

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* as a **System.Char**.

16 Exceptions

17
18

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

19
20
21

1 Convert.ToChar(System.UInt64) Method

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

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.Char**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.Char.MaxValue .

1 Convert.ToChar(System.String) Method

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

7 Summary

8 Converts a **System.String** to a **System.Char**.

9 Parameters

10
11

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

12
13
14

Return Value

15 *value* as a **System.Char**.

16 Exceptions

17
18

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

19
20
21

1 Convert.ToDateTime(System.DateTime)

2 Method

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

8 Summary

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

10 Parameters

11
12

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

13
14
15

14 Return Value

16 *value* is returned unchanged.

17 Description

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

19

1 Convert.ToDateTime(System.String)

2 Method

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

8 Summary

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

10 Parameters

11
12

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

13
14
15

Return Value

16 *value* as a **System.DateTime**.

17 Description

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

21 Exceptions

22
23

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

24
25
26

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

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

10 Summary

11 Converts a **System.String** to a **System.DateTime** structure.

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* as a **System.DateTime**.

19 Description

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

26 Exceptions

27
28

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

1
2
3

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

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

4 `[ILASM]`
5 `.method public hidebysig static decimal ToDecimal(int8`
6 `value)`

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

9 **Summary**

10 Converts a **System.SByte** to a **System.Decimal**.

11 **Type Attributes:**

- 12
 - CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Decimal**.

20 **Description**

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

23

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

3 Convert.ToDecimal(System.Byte) Method

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

9 Summary

10 Converts a **System.Byte** to a **System.Decimal**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Decimal**.

18

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

3 Convert.ToDecimal(System.Int16) Method

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

9 Summary

10 Converts a **System.Int16** to a **System.Decimal**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Decimal**.

18

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

3 Convert.ToDecimal(System.UInt16) 4 Method

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

10 Summary

11 Converts a **System.UInt16** to a **System.Decimal**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a **System.Decimal**.

21 Description

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

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

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

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

9 **Summary**

10 Converts a **System.Int32** to a **System.Decimal**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Decimal**.

18

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

3 Convert.ToDecimal(System.UInt32) 4 Method

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

10 Summary

11 Converts a **System.UInt32** to a **System.Decimal**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15
16

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

17
18
19

20 Return Value

20 *value* as a **System.Decimal**.

21 Description

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

24

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

3 Convert.ToDecimal(System.Int64) Method

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

9 Summary

10 Converts a **System.Int64** to a **System.Decimal**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Decimal**.

18

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

3 Convert.ToDecimal(System.UInt64) 4 Method

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

10 Summary

11 Converts a **System.UInt64** to a **System.Decimal**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15 Parameter	16 Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

17 Return Value

18 *value* as a **System.Decimal**.

19 Description

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

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

3 Convert.ToDecimal(System.Single) 4 Method

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

10 Summary

11 Converts a **System.Single** to a **System.Decimal**.

12 Parameters

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

16 Return Value

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

20 Exceptions

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

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

3 Convert.ToDecimal(System.Double) 4 Method

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

10 Summary

11 Converts a **System.Double** to a **System.Decimal**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

20 Exceptions

21
22

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

23
24
25

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

3 Convert.ToDecimal(System.String) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static decimal ToDecimal(string  
7 value)  
8 [C#]  
9 public static decimal ToDecimal(string value)
```

10 Summary

11 Converts a **System.String** to a **System.Decimal**.

12 Parameters

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

16 Return Value

18 *value* as a **System.Decimal**.

19 Description

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

23 Exceptions

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

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

3 Convert.ToDecimal(System.String, 4 System.IFormatProvider) Method

```
5 [ILASM]  
6 .method public hidebysig static decimal ToDecimal(string  
7 value, class System.IFormatProvider provider)  
  
8 [C#]  
9 public static decimal ToDecimal(string value,  
10 IFormatProvider provider)
```

11 Summary

12 Converts a **System.String** to a **System.Decimal**.

13 Parameters

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

17 Return Value

19 *value* as a **System.Decimal**.

20 Description

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

27 Exceptions

Exception	Condition
-----------	-----------

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

- 1
- 2
- 3

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

3 Convert.ToDecimal(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static decimal.ToDecimal(decimal  
7 value)  
8 [C#]  
9 public static decimal.ToDecimal(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.Decimal**.

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* is returned unchanged.

19 Description

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

21

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

3 Convert.ToDecimal(System.Boolean) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static decimal ToDecimal(bool  
7 value)  
8 [C#]  
9 public static decimal ToDecimal(bool value)
```

10 Summary

11 Converts a **System.Boolean** to a **System.Decimal**.

12 Parameters

13
14

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

15
16
17

16 Return Value

18 If *value* is **true** returns 1; if *value* is **false** returns 0.
19

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

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

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

9 **Summary**

10 Converts a **System.SByte** to a **System.Double**.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Double**.

20 **Description**

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

23

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

3 Convert.ToDouble(System.Byte) Method

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

9 Summary

10 Converts a **System.Byte** to a **System.Double**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Double**.

18

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

3 Convert.ToDouble(System.Int16) Method

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

9 Summary

10 Converts a **System.Int16** to a **System.Double**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Double**.

18

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

3 Convert.ToDouble(System.UInt16) 4 Method

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

10 Summary

11 Converts a **System.UInt16** to a **System.Double**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15
16

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

17
18
19

Return Value

20 *value* as a **System.Double**.

21 Description

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

24

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

3 Convert.ToDouble(System.Int32) Method

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

9 Summary

10 Converts a **System.Int32** to a **System.Double**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Double**.

18

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

3 Convert.ToDouble(System.UInt32) 4 Method

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

10 Summary

11 Converts a **System.UInt32** to a **System.Double**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a **System.Double**.

21 Description

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

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

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

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

9 **Summary**

10 Converts a **System.Int64** to a **System.Double**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Double**.

18

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

3 Convert.ToDouble(System.UInt64) 4 Method

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

10 Summary

11 Converts a **System.UInt64** to a **System.Double**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15
16

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

17
18
19

Return Value

20 *value* as a **System.Double**.

21 Description

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

24

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

3 Convert.ToDouble(System.Single) Method

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

9 Summary

10 Converts a **System.Single** to a **System.Double**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Double**.

18

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

3 Convert.ToDouble(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static float64 ToDouble(float64  
7 value)  
8 [C#]  
9 public static double ToDouble(double value)
```

10 Summary

11 Converts a **System.Double** to a **System.Double**.

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* is returned unchanged.

19 Description

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

21

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

3 Convert.ToDouble(System.Decimal) 4 Method

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

10 Summary

11 Converts a **System.Decimal** to a **System.Double**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19

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

3 Convert.ToDouble(System.String) Method

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

9 Summary

10 Converts a **System.String** to a **System.Double**.

11 Parameters

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

15 Return Value

17 *value* as a **System.Double**.

18 Description

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

22 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> 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
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Float System.Globalization.NumberStyles.AllowThousands style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

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

- 1
- 2
- 3

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

3 Convert.ToDouble(System.Boolean) 4 Method

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

10 Summary

11 Converts a **System.Boolean** to a **System.Double**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19

1 Convert.ToInt16(System.Boolean) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

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

15

1 Convert.ToInt16(System.Char) Method

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

7 Summary

8 Converts a **System.Char** to a **System.Int16**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16 Exceptions

17
18

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

19

20

21

1 Convert.ToInt16(System.SByte) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13
14
15

14 Return Value

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

17 Description

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

20

1 Convert.ToInt16(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.Int16**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToInt16(System.UInt16) Method

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

7 Summary

8 Converts a **System.UInt16** to a **System.Int16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

14 Return Value

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

17 Description

18 This member is not CLS-compliant. For a CLS-compliant alternative,
19 use **System.Convert.ToInt16(System.Int32)**.
20

21 Exceptions

22 Exception	23 Condition
System.OverflowException	<i>value</i> is greater than System.Int16.MaxValue . 24 25 26

1 Convert.ToInt16(System.Int32) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

12 Return Value

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

15 Exceptions

16
17

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

18
19
20

1 Convert.ToInt16(System.UInt32) Method

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

7 Summary

8 Converts a **System.UInt32** to a **System.Int16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21 Exceptions

22
23

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

24
25
26

1 Convert.ToInt16(System.Int16) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

14 *value* is returned unchanged.

15 Description

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

17

1 Convert.ToInt16(System.Int64) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

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

15 Exceptions

16
17

Exception	Condition
System.OverflowException	value is greater than System.Int16.MaxValue or less than System.Int16.MinValue .

18
19
20

1 Convert.ToInt16(System.UInt64) Method

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

7 Summary

8 Converts a **System.UInt64** to a **System.Int16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21 Exceptions

22
23

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

24
25
26

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

3 Convert.ToInt16(System.Single) Method

```
4 [ILASM]  
5 .method public hidebysig static int16 ToInt16(float32  
6 value)
```

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

9 Summary

10 Converts a **System.Single** to a **System.Int16**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

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

3 Convert.ToInt16(System.Double) Method

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

9 Summary

10 Converts a **System.Double** to a **System.Int16**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

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

3 Convert.ToInt16(System.Decimal) Method

```
4 [ILASM]  
5 .method public hidebysig static int16 ToInt16(decimal  
6 value)
```

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

9 Summary

10 Converts a **System.Decimal** to a **System.Int16**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

1 Convert.ToInt16(System.String) Method

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

6 Summary

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

8 Parameters

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

11 Return Value

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

13 Description

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

17 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> 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
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

value as a 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	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than

1
2
3

	System.Int16.MaxValue or less than System.Int16.MinValue.
--	--

1 Convert.ToInt32(System.Boolean) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

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

15

1 Convert.ToInt32(System.Char) Method

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

7 Summary

8 Converts a **System.Char** to a **System.Int32**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToInt32(System.SByte) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13

14 Return Value

15

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

17 Description

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

20

1 Convert.ToInt32(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.Int32**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToInt32(System.Int16) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

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

15

1 Convert.ToInt32(System.UInt16) Method

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

7 Summary

8 Converts a **System.UInt16** to a **System.Int32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21

1 Convert.ToInt32(System.UInt32) Method

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

7 Summary

8 Converts a **System.UInt32** to a **System.Int32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21 Exceptions

22
23

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

24
25
26

1 Convert.ToInt32(System.Int32) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

12 Return Value

14 *value* is returned unchanged.

15 Description

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

17

1 Convert.ToInt32(System.Int64) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

12 Return Value

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

15 Exceptions

16
17

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

18
19
20

1 Convert.ToInt32(System.UInt64) Method

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

7 Summary

8 Converts a **System.UInt64** to a **System.Int32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21 Exceptions

22
23

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

24
25
26

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

3 Convert.ToInt32(System.Single) Method

```
4 [ILASM]  
5 .method public hidebysig static int32 ToInt32(float32  
6 value)
```

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

9 Summary

10 Converts a **System.Single** to a **System.Int32**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

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

3 Convert.ToInt32(System.Double) Method

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

9 Summary

10 Converts a **System.Double** to a **System.Int32**.

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

22 Exceptions

23
24

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

25
26
27

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

3 Convert.ToInt32(System.Decimal) Method

```
4 [ILASM]  
5 .method public hidebysig static int32 ToInt32(decimal  
6 value)
```

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

9 Summary

10 Converts a **System.Decimal** to a **System.Int32**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

1 Convert.ToInt32(System.String) Method

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

6 Summary

7 Converts a **System.String** to a **System.Int32**.

8 Parameters

9
10

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

11
12
13

12 Return Value

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

15 Description

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

19 Exceptions

20
21

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

22
23
24

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

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

9 Summary

10 Converts a **System.String** to a **System.Int32**.

11 Parameters

12
13

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

14
15
16

Return Value

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

18 Description

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

25 Exceptions

26
27

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

1
2
3

	System.Int32.MaxValue or less than System.Int32.MinValue.
--	--

1 Convert.ToInt64(System.Boolean) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

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

15

1 Convert.ToInt64(System.Char) Method

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

7 Summary

8 Converts a **System.Char** to a **System.Int64**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToInt64(System.SByte) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13

14 Return Value

15

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

17 Description

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

20

1 Convert.ToInt64(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.Int64**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16

1 Convert.ToInt64(System.Int16) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

14 *value* as a 64-bit signed integer.
15

1 Convert.ToInt64(System.UInt16) Method

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

7 Summary

8 Converts a **System.UInt16** to a **System.Int64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21

1 Convert.ToInt64(System.Int32) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

Return Value

14 *value* as a 64-bit signed integer.
15

1 Convert.ToInt64(System.UInt32) Method

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

7 Summary

8 Converts a **System.UInt32** to a **System.Int64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21

1 Convert.ToInt64(System.UInt64) Method

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

7 Summary

8 Converts a **System.UInt64** to a **System.Int64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

14 Return Value

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

18 Description

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

21 Exceptions

22 Exception	23 Condition
System.OverflowException	<i>value</i> is greater than System.Int64.MaxValue .

24
25
26

1 Convert.ToInt64(System.Int64) Method

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

6 Summary

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

8 Parameters

9
10

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

11
12
13

12 Return Value

14 *value* is returned unchanged.

15 Description

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

17

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

3 Convert.ToInt64(System.Single) Method

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

9 Summary

10 Converts a **System.Single** to a **System.Int64**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

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

3 Convert.ToInt64(System.Double) Method

```
4 [ILASM]  
5 .method public hidebysig static int64 ToInt64(float64  
6 value)
```

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

9 Summary

10 Converts a **System.Double** to a **System.Int64**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

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

3 Convert.ToInt64(System.Decimal) Method

```
4 [ILASM]  
5 .method public hidebysig static int64 ToInt64(decimal  
6 value)
```

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

9 Summary

10 Converts a **System.Decimal** to a **System.Int64**.

11 Parameters

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

15 Return Value

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

18 Description

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

22 Exceptions

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

1 Convert.ToInt64(System.String) Method

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

6 Summary

7 Converts a **System.String** to a **System.Int64**.

8 Parameters

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

11 Return Value

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

13 Description

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

15 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> 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
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

value as a 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	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than

1
2
3

	System.Int64.MaxValue or less than System.Int64.MinValue.
--	--

1 Convert.ToSByte(System.Boolean) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13
14
15

14 Return Value

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

17 Description

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

20

1 Convert.ToSByte(System.SByte) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13

14 Return Value

15

16 *value* is returned unchanged.

17 Description

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

20

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

22

1 Convert.ToSByte(System.Char) Method

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

7 Summary

8 Converts a **System.Char** to a **System.SByte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

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

15 Return Value

17 *value* as a **System.SByte**.

18 Description

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

21 Exceptions

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

1 Convert.ToInt16(System.Int16) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13
14
15

14 Return Value

16 *value* as a **System.SByte**.

17 Description

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

20 Exceptions

21
22

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

23
24
25

1 Convert.ToSByte(System.UInt16) Method

```
2 [ILASM]  
3 .method public hidebysig static int8 ToSByte(unsigned int16  
4 value)  
  
5 [C#]  
6 public static sbyte ToSByte(ushort value)
```

7 Summary

8 Converts a **System.UInt16** to a **System.SByte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 16-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.SByte**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.SByte.MaxValue .

1 Convert.ToSByte(System.Int32) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13
14
15

14 Return Value

16 *value* as a **System.SByte**.

17 Description

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

20 Exceptions

21
22

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

23
24
25

1 Convert.ToSByte(System.UInt32) Method

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

7 Summary

8 Converts a **System.UInt32** to a **System.SByte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 32-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.SByte**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.SByte.MaxValue .

1 Convert.ToSByte(System.Int64) Method

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

6 Summary

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

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

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

14 Return Value

16 *value* as a **System.SByte**.

17 Description

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

20 Exceptions

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

1 Convert.ToSByte(System.UInt64) Method

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

7 Summary

8 Converts a **System.UInt64** to a **System.SByte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit unsigned integer value to be converted.

14 Return Value

15 *value* as a **System.SByte**.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> 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
<i>value</i>	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	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue .

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

Convert.ToSByte(System.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
<i>value</i>	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	<i>value</i> is greater than System.SByte.MaxValue or less than System.SByte.MinValue .

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

3 Convert.ToSByte(System.Decimal) Method

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

8 Summary

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

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

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

16 Return Value

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

19 Description

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

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

26 Exceptions

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

1 Convert.ToSByte(System.String) Method

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

6 Summary

7 Converts a **System.String** representation of a number to a
8 **System.SByte**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.

14 Return Value

15 *value* as a **System.SByte**.

16 Description

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

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

22 Exceptions

23 Exception	24 Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value in the specified format.
System.OverflowException	The numeric value of <i>value</i> is greater than System.SByte.MaxValue or less than

1
2
3

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

Return Value

value as a **System.SByte**.

Description

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

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

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

3 **Convert.ToSingle(System.SByte) Method**

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

9 **Summary**

10 Converts a **System.SByte** to a **System.Single**.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Single**.

20 **Description**

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

23

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

3 Convert.ToSingle(System.Byte) Method

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(unsigned  
6 int8 value)  
  
7 [C#]  
8 public static float ToSingle(byte value)
```

9 Summary

10 Converts a **System.Byte** to a **System.Single**.

11 Parameters

12
13

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

14

15 Return Value

16

17 *value* as a **System.Single**.

18

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

3 **Convert.ToIntSingle(System.Int16) Method**

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(int16  
6 value)  
7 [C#]  
8 public static float ToSingle(short value)
```

9 **Summary**

10 Converts a **System.Int16** to a **System.Single**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Single**.

18

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

3 **Convert.ToSingle(System.UInt16) Method**

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(unsigned  
6 int16 value)  
  
7 [C#]  
8 public static float ToSingle(ushort value)
```

9 **Summary**

10 Converts a **System.UInt16** to a **System.Single**.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Single**.

20 **Description**

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

23

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

3 **Convert.ToInt32(System.Int32) Method**

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(int32  
6 value)  
7 [C#]  
8 public static float ToSingle(int value)
```

9 **Summary**

10 Converts a **System.Int32** to a **System.Single**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Single**.

18

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

3 **Convert.ToSingle(System.UInt32) Method**

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(unsigned  
6 int32 value)  
  
7 [C#]  
8 public static float ToSingle(uint value)
```

9 **Summary**

10 Converts a **System.UInt32** to a **System.Single**.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Single**.

20 **Description**

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

23

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

3 **Convert.ToSingle(System.Int64) Method**

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

9 **Summary**

10 Converts a **System.Int64** to a **System.Single**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Single**.

18

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

3 **Convert.ToSingle(System.UInt64) Method**

```
4 [ILASM]  
5 .method public hidebysig static float32 ToSingle(unsigned  
6 int64 value)  
  
7 [C#]  
8 public static float ToSingle(ulong value)
```

9 **Summary**

10 Converts a **System.UInt64** to a **System.Single**.

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17
18

17 **Return Value**

19 *value* as a **System.Single**.

20 **Description**

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

23

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

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

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

9 **Summary**

10 Converts a **System.Single** to a **System.Single**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

17 *value* as a **System.Single**.

18 **Description**

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

20

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

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

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

9 **Summary**

10 Converts a **System.Double** to a **System.Single**.

11 **Parameters**

12
13

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

14
15
16

15 **Return Value**

17 *value* as a **System.Single**.

18 **Exceptions**

19
20

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

21
22
23

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

3 **Convert.ToSingle(System.Decimal)** 4 **Method**

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

10 **Summary**

11 Converts a **System.Decimal** to a **System.Single**.

12 **Parameters**

13
14

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

15
16
17

16 **Return Value**

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

19

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

3 Convert.ToSingle(System.String) Method

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

9 Summary

10 Converts a **System.String** to a **System.Single**.

11 Parameters

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

15 Return Value

17 *value* as a **System.Single**.

18 Description

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

22 Exceptions

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

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

3 Convert.ToSingle(System.String, 4 System.IFormatProvider) Method

```
5 [ILASM]  
6 .method public hidebysig static float32 ToSingle(string  
7 value, class System.IFormatProvider provider)  
  
8 [C#]  
9 public static float ToSingle(string value, IFormatProvider  
10 provider)
```

11 Summary

12 Converts a **System.String** to a **System.Single**.

13 Parameters

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

16 Return Value

17 *value* as a **System.Single**.
18

19 Description

20 This method parses *value* using the information in the
21 **System.Globalization.NumberFormatInfo** instance supplied by
22 *provider*. If *provider* is **null** or if a
23 **System.Globalization.NumberFormatInfo** cannot be obtained from
24 *provider*, the string is parsed using the formatting information of the
25 current system culture.
26

27 Exceptions

Exception	Condition
-----------	-----------

- 1
- 2
- 3

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

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

3 Convert.ToSingle(System.Boolean) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static float32 ToSingle(bool  
7 value)  
8 [C#]  
9 public static float ToSingle(bool value)
```

10 Summary

11 Converts a **System.Boolean** to a **System.Single**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19

1 Convert.ToString(System.Boolean) 2 Method

```
3 [ILASM]  
4 .method public hidebysig static string ToString(bool value)  
5 [C#]  
6 public static string ToString(bool value)
```

7 Summary

8 Converts a **System.Boolean** to a **System.String**.

9 Parameters

10
11

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

12
13
14

Return Value

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

16 Description

17 [Note: See **System.Boolean.ToString**.]
18

1 Convert.ToString(System.Char) Method

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

7 Summary

8 Converts a **System.Char** to a **System.String**.

9 Parameters

10
11

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

12

13 Return Value

14

15 *value* as a **System.String**.

16

1 Convert.ToString(System.SByte) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(int8 value)  
4 [C#]  
5 public static string ToString(sbyte value)
```

6 Summary

7 Converts a **System.SByte** to a **System.String**.

8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

10 Parameters

11
12

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

13

14 Return Value

15

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

17 Description

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

20

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

22

Convert.ToString(System.SByte, System.IFormatProvider) Method

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

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

Summary

Converts a **System.SByte** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

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

Return Value

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

Description

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

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

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

1 Convert.ToString(System.Byte) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(unsigned  
4 int8 value)  
5  
6 [C#]  
7 public static string ToString(byte value)
```

7 Summary

8 Converts a **System.Byte** to a **System.String**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16 Description

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

18

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

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**.]

1 Convert.ToString(System.Int16) Method

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

7 Summary

8 Converts a **System.Int16** to a **System.String**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16 Description

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

18

1 Convert.ToString(System.Int16, 2 System.IFormatProvider) Method

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

9 Summary

10 Converts a **System.Int16** to a **System.String**.

11 Parameters

Parameter	Description
<i>value</i>	The 16-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

14 Return Value

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

18 Description

19 This method converts *value* using the information in the
20 **System.Globalization.NumberFormatInfo** instance supplied by
21 *provider*. If *provider* is **null** or if a
22 **System.Globalization.NumberFormatInfo** cannot be obtained from
23 *provider*, the string is formatted in accordance with the current system
24 culture.

25 [Note: See **System.Int16.ToString**.]
26
27

1 Convert.ToString(System.UInt16) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(unsigned  
4 int16 value)  
  
5 [C#]  
6 public static string ToString(ushort value)
```

7 Summary

8 Converts a **System.UInt16** to a **System.String**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21
22

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

23

Convert.ToString(System.UInt16, System.IFormatProvider) Method

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

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

Summary

Converts a **System.UInt16** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 16-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

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

Description

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

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

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

1 Convert.ToString(System.Int32) Method

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

7 Summary

8 Converts a **System.Int32** to a **System.String**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16 Description

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

18

1 Convert.ToString(System.Int32, 2 System.IFormatProvider) Method

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

9 Summary

10 Converts a **System.Int32** to a **System.String**.

11 Parameters

12
13

Parameter	Description
<i>value</i>	The 32-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

14
15
16

Return Value

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

18 Description

19 This method converts *value* using the information in the
20 **System.Globalization.NumberFormatInfo** instance supplied by
21 *provider*. If *provider* is **null** or if a
22 **System.Globalization.NumberFormatInfo** cannot be obtained from
23 *provider*, the string is formatted in accordance with the current system
24 culture.

25
26 [Note: See **System.Int32.ToString**.]

27

1 Convert.ToString(System.UInt32) Method

```
2 [ILASM]
3 .method public hidebysig static string ToString(unsigned
4 int32 value)
5
6 [C#]
7 public static string ToString(uint value)
```

7 Summary

8 Converts a **System.UInt32** to a **System.String**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21
22

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

23

Convert.ToString(System.UInt32, System.IFormatProvider) Method

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

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

Summary

Converts a **System.UInt32** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 32-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

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

Description

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

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

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

1 Convert.ToString(System.Int64) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(int64  
4 value)  
5 [C#]  
6 public static string ToString(long value)
```

7 Summary

8 Converts a **System.Int64** to a **System.String**.

9 Parameters

10
11

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

12

13 Return Value

14

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

16 Description

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

18

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
<i>value</i>	The 64-bit signed integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

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**.]

1 Convert.ToString(System.UInt64) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(unsigned  
4 int64 value)  
  
5 [C#]  
6 public static string ToString(ulong value)
```

7 Summary

8 Converts a **System.UInt64** to a **System.String**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

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

18 Description

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

21
22

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

23

Convert.ToString(System.UInt64, System.IFormatProvider) Method

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

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

Summary

Converts a **System.UInt64** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 64-bit unsigned integer value to be converted.
<i>provider</i>	A System.IFormatProvider that supplies a System.Globalization.NumberFormatInfo containing culture-specific formatting information.

Return Value

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

Description

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

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

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

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

3 Convert.ToString(System.Single) Method

```
4 [ILASM]  
5 .method public hidebysig static string ToString(float32  
6 value)  
7 [C#]  
8 public static string ToString(float value)
```

9 Summary

10 Converts a **System.Single** to a **System.String**.

11 Parameters

12
13

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

14

15 Return Value

16

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

18 Description

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

20

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

3 Convert.ToString(System.Single, 4 System.IFormatProvider) Method

```
5 [ILASM]  
6 .method public hidebysig static string ToString(float32  
7 value, class System.IFormatProvider provider)  
  
8 [C#]  
9 public static string ToString(float value, IFormatProvider  
10 provider)
```

11 Summary

12 Converts a **System.Single** to a **System.String**.

13 Parameters

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

17 Return Value

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

20 Description

21 This method converts *value* using the information in the
22 **System.Globalization.NumberFormatInfo** instance supplied by
23 *provider*. If *provider* is **null** or if a
24 **System.Globalization.NumberFormatInfo** cannot be obtained from
25 *provider*, the string is formatted in accordance with the current system
26 culture.

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

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

3 **Convert.ToString(System.Double) Method**

```
4 [ILASM]  
5 .method public hidebysig static string ToString(float64  
6 value)  
7 [C#]  
8 public static string ToString(double value)
```

9 **Summary**

10 Converts a **System.Double** to a **System.String**.

11 **Parameters**

12
13

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

14

15 **Return Value**

16

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

18 **Description**

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

20

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

3 Convert.ToString(System.Double, 4 System.IFormatProvider) Method

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

11 Summary

12 Converts a **System.Double** to a **System.String**.

13 Parameters

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

17 Return Value

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

20 Description

21 This method converts *value* using the information in the
22 **System.Globalization.NumberFormatInfo** instance supplied by
23 *provider*. If *provider* is **null** or if a
24 **System.Globalization.NumberFormatInfo** cannot be obtained from
25 *provider*, the string is formatted in accordance with the current system
26 culture.

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

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

3 Convert.ToString(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static string ToString(decimal  
7 value)  
8 [C#]  
9 public static string ToString(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.String**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19 Description

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

21

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

3 Convert.ToString(System.Decimal, 4 System.IFormatProvider) Method

```
5 [ILASM]  
6 .method public hidebysig static string ToString(decimal  
7 value, class System.IFormatProvider provider)  
  
8 [C#]  
9 public static string ToString(decimal value,  
10 IFormatProvider provider)
```

11 Summary

12 Converts a **System.Decimal** to a **System.String**.

13 Parameters

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

17 Return Value

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

20 Description

21 This method converts *value* using the information in the
22 **System.Globalization.NumberFormatInfo** instance supplied by
23 *provider*. If *provider* is **null** or if a
24 **System.Globalization.NumberFormatInfo** cannot be obtained from
25 *provider*, the string is formatted in accordance with the current system
26 culture.

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

1 Convert.ToString(System.DateTime)

2 Method

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

8 Summary

9 Converts a **System.DateTime** value to a **System.String**.

10 Parameters

11
12

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

13
14
15

14 Return Value

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

17 Description

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

19

1 Convert.ToString(System.DateTime, 2 System.IFormatProvider) Method

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

10 Summary

11 Converts a **System.DateTime** value to a **System.String**.

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19 Description

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

26
27

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

28

1 Convert.ToString(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static string ToString(string  
4 value)  
5 [C#]  
6 public static string ToString(string value)
```

7 Summary

8 Returns the specified string.

9 Parameters

10
11

Parameter	Description
<i>value</i>	A System.String .

12

13 Return Value

14

15 *value* is returned unchanged.

16 Description

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

18

1 Convert.ToInt16(System.Boolean)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int16  
5 ToUInt16(bool value)  
  
6 [C#]  
7 public static ushort ToUInt16(bool value)
```

8 Summary

9 Converts a **System.Boolean** to a **System.UInt16**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

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

19 Description

20 This member is not CLS-compliant. For a CLS-compliant alternative,
21 use **System.Convert.ToInt32(System.Boolean)**.

22

1 Convert.ToInt16(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int16  
4 ToInt16(valuetype System.Char value)  
  
5 [C#]  
6 public static ushort ToInt16(char value)
```

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 16-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToInt32(System.Char)**.

21

1 Convert.ToInt16(System.SByte) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int16  
4 ToInt16(int8 value)  
  
5 [C#]  
6 public static ushort ToInt16(sbyte value)
```

7 Summary

8 Converts a **System.SByte** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The System.SByte value to be converted.

14 Return Value

15 *value* as a 16-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt16.MinValue .

1 Convert.ToInt16(System.Byte) Method

```
2 [ILASM]
3 .method public hidebysig static unsigned int16
4 ToInt16(unsigned int8 value)
5
6 [C#]
7 public static ushort ToInt16(byte value)
```

7 Summary

8 Converts a **System.Byte** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 16-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToInt32(System.Byte)**.

21

1 Convert.ToInt16(System.Int16) Method

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

7 Summary

8 Converts a **System.Int16** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 16-bit unsigned integer.

18 Description

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

21 Exceptions

22
23

Exception	Condition
System.OverflowException	<i>value</i> is less than System.UInt16.MinValue .

24
25
26

1 Convert.ToInt16(System.Int32) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int16  
4 ToInt16(int32 value)  
  
5 [C#]  
6 public static ushort ToInt16(int value)
```

7 Summary

8 Converts a **System.Int32** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 32-bit signed integer value to be converted.

14 Return Value

15 value as a 16-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .

1 Convert.ToInt16(System.UInt16)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int16  
5 ToUInt16(unsigned int16 value)  
  
6 [C#]  
7 public static ushort ToUInt16(ushort value)
```

8 Summary

9 Converts a **System.UInt16** to a **System.UInt16**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* is returned unchanged.

19 Description

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

21
22
23

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

24

1 Convert.ToInt16(System.UInt32)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int16  
5 ToUInt16(unsigned int32 value)  
  
6 [C#]  
7 public static ushort ToUInt16(uint value)
```

8 Summary

9 Converts a **System.UInt32** to a **System.UInt16**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* as a 16-bit unsigned integer.

19 Description

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

22 Exceptions

23
24

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue .

25
26
27

1 Convert.ToInt16(System.Int64) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int16  
4 ToInt16(int64 value)  
  
5 [C#]  
6 public static ushort ToInt16(long value)
```

7 Summary

8 Converts a **System.Int64** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit signed integer value to be converted.

14 Return Value

15 *value* as a 16-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .

1 Convert.ToInt16(System.UInt64)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int16  
5 ToUInt16(unsigned int64 value)  
  
6 [C#]  
7 public static ushort ToUInt16(ulong value)
```

8 Summary

9 Converts a **System.UInt64** to a **System.UInt16**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* as a 16-bit unsigned integer.

19 Description

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

22 Exceptions

23
24

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue .

25
26
27

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

Convert.ToInt16(System.Single) Method

```
[ILASM]
.method public hidebysig static unsigned int16
ToUInt16(float32 value)

[C#]
public static ushort ToUInt16(float value)
```

Summary

Converts a **System.Single** to a **System.UInt16**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 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	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .

1
2
3

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

3 Convert.ToInt16(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int16  
7 ToUInt16(float64 value)  
  
8 [C#]  
9 public static ushort ToUInt16(double value)
```

10 Summary

11 Converts a **System.Double** to a **System.UInt16**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 16-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

27 This member is not CLS-compliant. For a CLS-compliant alternative,
28 use **System.Convert.ToInt32(System.Double)**.

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .
---------------------------------	---

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

3 Convert.ToInt16(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int16  
7 ToInt16(decimal value)  
  
8 [C#]  
9 public static ushort ToInt16(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.UInt16**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 16-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

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

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .
---------------------------------	---

1 Convert.ToInt16(System.String) Method

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

7 Summary

8 Converts a **System.String** to a **System.UInt16**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

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

15 Return Value

17 *value* as a 16-bit unsigned integer.

18 Description

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

23 This member is not CLS-compliant. For a CLS-compliant alternative,
24 use **System.Convert.ToInt32(System.String)**.

25 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .

1
2
3

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

```
3 [ILASM]  
4 .method public hidebysig static unsigned int16  
5 ToInt16(string value, class System.IFormatProvider  
6 provider)  
  
7 [C#]  
8 public static ushort ToInt16(string value, IFormatProvider  
9 provider)
```

10 Summary

11 Converts a **System.String** to a **System.UInt16**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

17 Return Value

18 *value* as a 16-bit unsigned integer.

19 Description

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

26 This member is not CLS-compliant. For a CLS-compliant alternative,
27 use **System.Convert.ToInt32(System.String,
28 System.IFormatProvider)**.

1 **Exceptions**
2
3

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt16.MaxValue or less than System.UInt16.MinValue .

4
5
6

1 Convert.ToInt32(System.Boolean)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int32  
5 ToInt32(bool value)  
  
6 [C#]  
7 public static uint ToInt32(bool value)
```

8 Summary

9 Converts a **System.Boolean** to a **System.UInt32**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13

14

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

15

16 Return Value

17

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

19 Description

20 This member is not CLS-compliant. For a CLS-compliant alternative,
21 use **System.Convert.ToInt64(System.Boolean)**.

22

1 Convert.ToInt32(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int32  
4 ToInt32(valuetype System.Char value)  
5  
6 [C#]  
7 public static uint ToInt32(char value)
```

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 32-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToInt64(System.Char)**.

21

1 Convert.ToInt32(System.SByte) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int32  
4 ToInt32(int8 value)  
  
5 [C#]  
6 public static uint ToInt32(sbyte value)
```

7 Summary

8 Converts a **System.SByte** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 32-bit unsigned integer.

18 Description

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

21 Exceptions

22
23

Exception	Condition
System.OverflowException	<i>value</i> is less than System.UInt32.MinValue .

24
25
26

1 Convert.ToInt32(System.Byte) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int32  
4 ToInt32(unsigned int8 value)  
5  
6 [C#]  
7 public static uint ToInt32(byte value)
```

7 Summary

8 Converts a **System.Byte** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 32-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToInt64(System.Byte)**.

21

1 Convert.ToInt32(System.Int16) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int32  
4 ToInt32(int16 value)  
  
5 [C#]  
6 public static uint ToInt32(short value)
```

7 Summary

8 Converts a **System.Int16** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 16-bit signed integer value to be converted.

14 Return Value

15 *value* as a 32-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt32.MinValue .

1 Convert.ToInt32(System.UInt16)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int32  
5 ToUInt32(unsigned int16 value)  
  
6 [C#]  
7 public static uint ToUInt32(ushort value)
```

8 Summary

9 Converts a **System.UInt16** to a **System.UInt32**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13

14

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

15

16 Return Value

17

18 *value* as a 32-bit unsigned integer.

19 Description

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

22

1 Convert.ToInt32(System.Int32) Method

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

7 Summary

8 Converts a **System.Int32** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 32-bit signed integer value to be converted.

14 Return Value

15 *value* as a 32-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt32.MinValue .

1 Convert.ToInt32(System.UInt32)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int32  
5 ToUInt32(unsigned int32 value)  
  
6 [C#]  
7 public static uint ToUInt32(uint value)
```

8 Summary

9 Converts a **System.UInt32** to a **System.UInt32**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* is returned unchanged.

19 Description

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

21
22 This member is not CLS-compliant. For a CLS-compliant alternative,
23 use **System.Convert.ToInt64(System.Int64)**.

24

1 Convert.ToInt32(System.Int64) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int32  
4 ToInt32(int64 value)  
  
5 [C#]  
6 public static uint ToInt32(long value)
```

7 Summary

8 Converts a **System.Int64** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit signed integer value to be converted.

14 Return Value

15 *value* as a 32-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .

1 Convert.ToInt32(System.UInt64)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int32  
5 ToUInt32(unsigned int64 value)  
  
6 [C#]  
7 public static uint ToUInt32(ulong value)
```

8 Summary

9 Converts a **System.UInt64** to a **System.UInt32**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* as a 32-bit unsigned integer.

19 Description

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

22 Exceptions

23
24

Exception	Condition
System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue .

25
26
27

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

Convert.ToInt32(System.Single) Method

```
[ILASM]
.method public hidebysig static unsigned int32
ToUInt32(float32 value)

[C#]
public static uint ToUInt32(float value)
```

Summary

Converts a **System.Single** to a **System.UInt32**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 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	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .

1
2
3

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

3 Convert.ToInt32(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int32  
7 ToUInt32(float64 value)  
  
8 [C#]  
9 public static uint ToUInt32(double value)
```

10 Summary

11 Converts a **System.Double** to a **System.UInt32**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 32-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

27 This member is not CLS-compliant. For a CLS-compliant alternative,
28 use **System.Convert.ToInt64(System.Double)**.

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .
---------------------------------	---

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

3 Convert.ToInt32(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int32  
7 ToInt32(decimal value)  
8 [C#]  
9 public static uint ToInt32(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.UInt32**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 32-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

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

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .
---------------------------------	---

1 Convert.ToInt32(System.String) Method

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

7 Summary

8 Converts a **System.String** to a **System.UInt32**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

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

15 Return Value

17 *value* as a 32-bit unsigned integer.

18 Description

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

23 This member is not CLS-compliant. For a CLS-compliant alternative,
24 use **System.Convert.ToInt64(System.String)**.

25 Exceptions

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .

1
2
3

Convert.ToInt32(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
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	An object that implements the System.IFormatProvider interface and supplies a System.Globalization.NumberFormatInfo instance containing culture-specific formatting information.

Return Value

value as an 32-bit unsigned integer.

Description

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

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Convert.ToInt64(System.String, System.IFormatProvider)**.

1 **Exceptions**
2
3

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt32.MaxValue or less than System.UInt32.MinValue .

4
5
6

1 Convert.ToInt64(System.Boolean) 2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int64  
5 ToUInt64(bool value)  
6  
7 [C#]  
8 public static ulong ToUInt64(bool value)
```

8 Summary

9 Converts a **System.Boolean** to a **System.UInt64**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13

14

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

15

16 Return Value

17

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

19 Description

20 This member is not CLS-compliant. For a CLS-compliant alternative,
21 use **System.Convert.ToDecimal(System.Boolean)**.

22

1 Convert.ToInt64(System.Char) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int64  
4 ToInt64(valuetype System.Char value)  
5  
6 [C#]  
7 public static ulong ToInt64(char value)
```

7 Summary

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

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 64-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToDecimal(System.Char)**.

21

1 Convert.ToInt64(System.SByte) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int64  
4 ToInt64(int8 value)  
  
5 [C#]  
6 public static ulong ToInt64(sbyte value)
```

7 Summary

8 Converts a **System.SByte** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 64-bit unsigned integer.

18 Description

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

21 Exceptions

22
23

Exception	Condition
System.OverflowException	<i>value</i> is less than System.UInt64.MinValue .

24
25
26

1 Convert.ToInt64(System.Byte) Method

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

7 Summary

8 Converts a **System.Byte** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

15 Return Value

17 *value* as a 64-bit unsigned integer.

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Convert.ToDecimal(System.Byte)**.

21

1 Convert.ToInt64(System.Int16) Method

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

7 Summary

8 Converts a **System.Int16** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 16-bit signed integer value to be converted.

14 Return Value

15 *value* as a 64-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt64.MinValue .

1 Convert.ToInt64(System.UInt16)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int64  
5 ToUInt64(unsigned int16 value)  
  
6 [C#]  
7 public static ulong ToUInt64(ushort value)
```

8 Summary

9 Converts a **System.UInt16** to a **System.UInt64**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* as a 64-bit unsigned integer.

19 Description

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

22

1 Convert.ToInt64(System.Int32) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int64  
4 ToInt64(int32 value)  
  
5 [C#]  
6 public static ulong ToInt64(int value)
```

7 Summary

8 Converts a **System.Int32** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 32-bit signed integer value to be converted.

14 Return Value

15 *value* as a 64-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt64.MinValue .

1 Convert.ToInt64(System.UInt32)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int64  
5 ToUInt64(unsigned int32 value)  
  
6 [C#]  
7 public static ulong ToUInt64(uint value)
```

8 Summary

9 Converts a **System.UInt32** to a **System.UInt64**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13

14

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

15

16 Return Value

17

18 *value* as a 64-bit unsigned integer.

19 Description

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

22

1 Convert.ToInt64(System.Int64) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int64  
4 ToUInt64(int64 value)  
  
5 [C#]  
6 public static ulong ToUInt64(long value)
```

7 Summary

8 Converts a **System.Int64** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The 64-bit signed integer value to be converted.

14 Return Value

15 *value* as a 64-bit unsigned integer.

16 Description

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

19 Exceptions

20 Exception	21 Condition
System.OverflowException	<i>value</i> is less than System.UInt64.MinValue .

1 Convert.ToInt64(System.UInt64)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static unsigned int64  
5 ToUInt64(unsigned int64 value)  
  
6 [C#]  
7 public static ulong ToUInt64(ulong value)
```

8 Summary

9 Converts a **System.UInt64** to a **System.UInt64**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16
17

16 Return Value

18 *value* is returned unchanged.

19 Description

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

21
22
23

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

24

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

Convert.ToInt64(System.Single) Method

```
[ILASM]
.method public hidebysig static unsigned int64
ToUInt64(float32 value)

[C#]
public static ulong ToUInt64(float value)
```

Summary

Converts a **System.Single** to a **System.UInt64**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The 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	<i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue .

1
2
3

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

3 Convert.ToInt64(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int64  
7 ToInt64(float64 value)  
8 [C#]  
9 public static ulong ToInt64(double value)
```

10 Summary

11 Converts a **System.Double** to a **System.UInt64**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 64-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

27 This member is not CLS-compliant. For a CLS-compliant alternative,
28 use **System.Convert.ToDecimal(System.Double)**.

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue .
---------------------------------	---

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

3 Convert.ToInt64(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static unsigned int64  
7 ToInt64(decimal value)  
8 [C#]  
9 public static ulong ToInt64(decimal value)
```

10 Summary

11 Converts a **System.Decimal** to a **System.UInt64**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

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

18 Return Value

20 *value* as a 64-bit unsigned integer. *value* is rounded prior to
21 conversion.

22 Description

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

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

29 Exceptions

Exception	Condition
-----------	-----------

1
2
3

System.OverflowException	<i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue .
---------------------------------	---

1 Convert.ToInt64(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static unsigned int64  
4 ToUInt64(string value)  
  
5 [C#]  
6 public static ulong ToUInt64(string value)
```

7 Summary

8 Converts a **System.String** to a **System.UInt64**.

9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

11 Parameters

12 Parameter	13 Description
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.

14 Return Value

15 *value* as a 64-bit unsigned integer.

16 Description

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

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

19 Exceptions

20 Exception	21 Condition
System.ArgumentNullException	<i>value</i> is null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue .

1
2
3

Convert.ToInt64(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
<i>value</i>	The System.String to be converted. The string is in the System.Globalization.NumberStyles.Integer style.
<i>provider</i>	A System.Object that implements the System.IFormatProvider interface and supplies a System.Globalization.NumberFormatInfo instance containing culture-specific formatting information.

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)**.

1 **Exceptions**
2
3

Exception	Condition
System.ArgumentNullException	<i>value</i> is a null reference.
System.FormatException	<i>value</i> cannot be converted to a numeric value.
System.OverflowException	The numeric value of <i>value</i> is greater than System.UInt64.MaxValue or less than System.UInt64.MinValue .

4
5