

System.Xml.XmlConvert Class

```
[ILASM]
.class public XmlConvert extends System.Object

[C#]
public class XmlConvert
```

Assembly Info:

- Name: System.Xml
- 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

Encodes and decodes XML names and provides methods for converting between common language infrastructure (CLI) types and XML Schema Definition language (XSD) types. When converting data types, the values returned are locale independent.

Inherits From: System.Object

Library: XML

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

Description

Element and attribute names or ID values are limited to a range of XML characters according to the Extensible Markup Language (XML) 1.0 (Second Edition) recommendation, located at www.w3.org/TR/2000/REC-xml-20001006.html. When names contain invalid characters, they need to be translated into valid XML names.

Many languages and applications allow Unicode characters in their names, which are not valid in XML names. For example, if 'Order Detail' were a column heading in a database, the database allows the space between the words Order and Detail. However, in XML, the space between Order and Detail is considered an invalid XML character. Thus, the space, the invalid character, needs to be converted into an escaped hexadecimal encoding and can be decoded later.

The **System.Xml.XmlConvert.EncodeName** and

System.Xml.XmlConvert.DecodeName methods are used to translate invalid XML names into valid XML names and vice versa.

System.Xml.XmlConvert provides methods that enable the conversion of a **System.String** to a CLI data type and vice-versa. Locale settings are not taken into account during data conversion.

System.Xml.XmlConvert also provides methods that convert between XML Schema Definition (XSD) data types (see <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>) and their corresponding common language infrastructure (CLI) data types. The following table shows the XSD data types and their corresponding CLI data types.

XSD data type	CLI data type
hexBinary	A System.Byte array
base64Binary	A System.Byte array
Boolean	System.Boolean
Byte	System.SByte
normalizedString	System.String
Date	System.DateTime
duration	System.TimeSpan
dateTime	System.DateTime
decimal	System.Decimal
Double	System.Double
ENTITIES	A System.String array
ENTITY	System.String
Float	System.Single
gMonthDay	System.DateTime
gDay	System.DateTime
gYear	System.DateTime
gYearMonth	System.DateTime
ID	System.String
IDREF	System.String
IDREFS	A System.String array
int	System.Int32
integer	System.Decimal
language	System.String
long	System.Int64
month	System.DateTime
Name	System.String
NCName	System.String

negativeInteger	System.Decimal
NMTOKEN	System.String
NMTOKENS	A System.String array
nonNegativeInteger	System.Decimal
nonPositiveInteger	System.Decimal
NOTATION	System.String
positiveInteger	System.Decimal
short	System.Int16
string	System.String
time	System.DateTime
timePeriod	System.DateTime
token	System.String
unsignedByte	System.Byte
unsignedInt	System.UInt32
unsignedLong	System.UInt64
unsignedShort	System.UInt16
anyURI	System.Uri

1

2

1 XmlConvert() Constructor

```
2 [ILASM]  
3 family specialname instance void .ctor()  
4  
5 [C#]  
6 protected XmlConvert()
```

6 Summary

7 Constructs a new instance of the **System.Xml.XmlConvert** class.

8

1 XmlConvert.DecodeName(System.String)

2 Method

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

8 Summary

9 Decodes a name.

10 Parameters

Parameter	Description
<i>name</i>	A System.String specifying the name to be decoded.

14 Return Value

16 A **System.String** containing the decoded name.

17 Description

18 Names are decoded using the following rules:

- 19 • Names are decoded from left to right.
- 20 • Any sequence `_xHHHH_` (where HHHH stands for a valid, four
21 digit hexadecimal UCS-2 code) that has not been previously
22 decoded is transformed into the corresponding Unicode 2.1
23 (Unicode 3.0 if supported by the application) character.
- 24 • No short forms are recognized. They are passed on without
25 translation. For example, `"_x70_"` or `"__"` are not decoded.

26 [Note: This method does the reverse of the
27 **System.Xml.XmlConvert.EncodeName**,
28 **System.Xml.XmlConvert.EncodeLocalName**, and
29 **System.Xml.XmlConvert.EncodeNmToken** methods.]

30 Example

1 The following example demonstrates the valid and invalid character
2 formats for decoding.

```
3  
4       [C#]  
  
5       using System;  
6       using System.Xml;  
7  
8       public class App {  
9  
10          public static void Main() {  
11  
12             Console.WriteLine("{0}: {1}: {2}",  
13                 // _x0069_ decodes to i  
14                 XmlConvert.DecodeName("Order #1_x0069_"),  
15  
16                 // missing beginning _  
17                 XmlConvert.DecodeName("Order #1x0069_"),  
18  
19                 // short form  
20                 XmlConvert.DecodeName("Order #1_x69_"));  
21          }  
22       }  
23
```

24 The output is

25
26 Order #1i: Order #1x0069_: Order #1_x69_

27

1 XmlConvert.EncodeLocalName(System.String) Method

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

8 Summary

9 Converts a name to a valid XML local name.

10 Parameters

11
12

Parameter	Description
<i>name</i>	A System.String specifying the name to be encoded.

13
14
15

14 Return Value

16 A **System.String** containing the XML local name. If *name* is **null** or
17 **System.String.Empty**, *name* is returned.

18 Description

19 This method is similar to the **System.Xml.XmlConvert.EncodeName**
20 method except that it encodes the colon (:) character, which
21 guarantees that the name can be used as the local name part of a
22 namespace qualified name.

23 Example

24

25 The following example compares the
26 **System.Xml.XmlConvert.EncodeLocalName**,
27 **System.Xml.XmlConvert.EncodeName**, and
28 **System.Xml.XmlConvert.EncodeNmToken** methods when the
29 name to be encoded is "7: +".

30
31

```
32 using System;
33 using System.Xml;
34
35 public class App {
36
37     public static void Main() {
```

```

1
2         Console.WriteLine("LocalName  {0}",
3             XmlConvert.EncodeLocalName("7:+"));
4         Console.WriteLine("Name  {0}",
5             XmlConvert.EncodeName("7:+"));
6         Console.WriteLine("NmToken  {0}",
7             XmlConvert.EncodeNmToken("7:+"));
8     }
9 }

```

```

10 The output is
11
12 LocalName _x0037__x003A__x002B_
13
14 Name _x0037_:_x002B_
15
16 NmToken 7:_x002B_

```

17

XmlConvert.EncodeName(System.String)

Method

```
[ILASM]
.method public hidebysig static string EncodeName(string
name)

[C#]
public static string EncodeName(string name)
```

Summary

Converts a name to a valid XML name.

Parameters

Parameter	Description
<i>name</i>	A System.String specifying the name to be encoded.

Return Value

A **System.String** containing the XML name. If *name* is **null** or **System.String.Empty**, *name* is returned.

Description

This method translates invalid characters, such as spaces or half-width Katakana, that need to be mapped to XML names without the support or presence of schemas. The invalid characters are translated into escaped numeric entity encodings.

The escape character is '_'. Any XML name character that does not conform to the W3C Extensible Markup Language (XML) 1.0 specification is escaped as `_xHHHH_`. The `HHHH` string stands for the four-digit hexadecimal UCS-2 code for the character in most significant bit first order. For example, the name "Order Details" is encoded as "Order_x0020_Details".

The underscore character does not need to be escaped unless it is followed by a character sequence that together with the underscore can be misinterpreted as an escape sequence when decoding the name. No short forms are encoded. For example, the forms `"_x20_"` and `"__"` are not encoded.

This method guarantees the name is valid according to the XML specification. It allows colons in any position, which means the name may still be invalid according to the W3C Namespace Specification

1 (www.w3.org/TR/REC-xml-names). To guarantee it is a valid
2 namespace qualified name use the
3 **System.Xml.XmlConvert.EncodeLocalName** method for the prefix
4 and local name parts and join the result with a colon.

5 **Example**

6

7 See the **System.Xml.XmlConvert.EncodeLocalName** method for an
8 example comparing the
9 **System.Xml.XmlConvert.EncodeLocalName**,
10 **System.Xml.XmlConvert.EncodeName**, and
11 **System.Xml.XmlConvert.EncodeNmToken** methods.

12

1 XmlConvert.EncodeNmToken(System.String) Method

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

8 Summary

9 Converts a name to a valid XML name token.

10 Parameters

11
12

Parameter	Description
<i>name</i>	A System.String specifying the name to be encoded.

13
14
15

Return Value

16 A **System.String** containing the XML name token. If *name* is **null** or
17 **System.String.Empty**, *name* is returned.

18 Example

19

20 See the **System.Xml.XmlConvert.EncodeLocalName** method for an
21 example comparing the
22 **System.Xml.XmlConvert.EncodeLocalName**,
23 **System.Xml.XmlConvert.EncodeName**, and
24 **System.Xml.XmlConvert.EncodeNmToken** methods.

25

XmlConvert.ToBoolean(System.String)

Method

```
[ILASM]
.method public hidebysig static bool ToBoolean(string s)

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

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Boolean** equivalent of s.

Description

This method removes leading and trailing white space. After this trimming, valid strings are "1" and "true", which return **true**, and "0" and "false", which return **false**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s does not represent a System.Boolean value.

XmlConvert.ToByte(System.String)

Method

```
[ILASM]
.method public hidebysig static unsigned int8 ToByte(string
s)

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

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Byte** equivalent of s.

Description

This method calls **System.Byte.Parse(s, System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Byte.MinValue or greater than System.Byte.MaxValue .

XmlConvert.ToChar(System.String)

Method

```
[ILASM]
.method public hidebysig static valuetype System.Char
ToChar(string s)

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

Summary

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

Parameters

Parameter	Description
s	The string containing a single character to convert.

Return Value

The **System.Char** equivalent of s.

Description

This method calls **System.Char.Parse(s)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s contains more than one character.

1 XmlConvert.ToDateTime(System.String)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
s	The System.String to convert.

13
14
15

Return Value

16 The **System.DateTime** equivalent of s.

17 Description

18 s is required to be in one of the following string formats or a
19 **System.FormatException** is thrown.

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

"yyyy-MM-ddTHH:mm:ss"

"yyyy-MM-ddTHH:mm:ss.f"

"yyyy-MM-ddTHH:mm:ss.ff"

"yyyy-MM-ddTHH:mm:ss.fff"

"yyyy-MM-ddTHH:mm:ss.ffff"

"yyyy-MM-ddTHH:mm:ss.fffff"

"yyyy-MM-ddTHH:mm:ss.ffffff"

"yyyy-MM-ddTHH:mm:ss.fffffff"

"yyyy-MM-ddTHH:mm:ssZ"

1 "yyyy-MM-ddTHH:mm:ss.fZ"
2
3 "yyyy-MM-ddTHH:mm:ss.ffZ"
4
5 "yyyy-MM-ddTHH:mm:ss.fffZ"
6
7 "yyyy-MM-ddTHH:mm:ss.ffffZ"
8
9 "yyyy-MM-ddTHH:mm:ss.fffffZ"
10
11 "yyyy-MM-ddTHH:mm:ss.fffffZ"
12
13 "yyyy-MM-ddTHH:mm:ss.fffffZ"
14
15 "yyyy-MM-ddTHH:mm:sszzzzzz"
16
17 "yyyy-MM-ddTHH:mm:ss.fzzzzzz"
18
19 "yyyy-MM-ddTHH:mm:ss.ffzzzzzz"
20
21 "yyyy-MM-ddTHH:mm:ss.fffzzzzzz"
22
23 "yyyy-MM-ddTHH:mm:ss.ffffzzzzzz"
24
25 "yyyy-MM-ddTHH:mm:ss.fffffzzzzzz"
26
27 "yyyy-MM-ddTHH:mm:ss.fffffzzzzzz"
28
29 "yyyy-MM-ddTHH:mm:ss.fffffzzzzzz"
30
31 "HH:mm:ss"
32
33 "HH:mm:ss.f"
34
35 "HH:mm:ss.ff"
36
37 "HH:mm:ss.fff"
38
39 "HH:mm:ss.ffff"
40
41 "HH:mm:ss.fffff"
42
43 "HH:mm:ss.fffffZ"
44
45 "HH:mm:ss.fffffZ"
46
47 "HH:mm:ssZ"
48
49 "HH:mm:ss.fZ"
50
51 "HH:mm:ss.ffZ"
52
53 "HH:mm:ss.fffZ"

1	
2	"HH:mm:ss.ffffZ"
3	
4	"HH:mm:ss.fffffZ"
5	
6	"HH:mm:ss.ffffffZ"
7	
8	"HH:mm:ss.fffffffZ"
9	
10	"HH:mm:sszzzzzz"
11	
12	"HH:mm:ss.fzzzzzz"
13	
14	"HH:mm:ss.ffzzzzzz"
15	
16	"HH:mm:ss.fffzzzzzz"
17	
18	"HH:mm:ss.ffffzzzzzz"
19	
20	"HH:mm:ss.fffffzzzzzz"
21	
22	"HH:mm:ss.ffffffzzzzzz"
23	
24	"HH:mm:ss.fffffffzzzzzz"
25	
26	"yyyy-MM-dd"
27	
28	"yyyy-MM-ddZ"
29	
30	"yyyy-MM-ddzzzzzz"
31	
32	"yyyy-MM"
33	
34	"yyyy-MMZ"
35	
36	"yyyy-MMzzzzzz"
37	
38	"yyyy"
39	
40	"yyyyZ"
41	
42	"yyyyzzzzzz"
43	
44	"--MM-dd"
45	
46	"--MM-ddZ"
47	
48	"--MM-ddzzzzzz"
49	
50	"---dd"
51	
52	"---ddZ"
53	

1 "---ddzzzzzz"
2
3 "--MM--"
4
5 "--MM--Z"
6
7 "--MM--zzzzzz"

8 **Exceptions**

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is System.String.Empty or is not in the correct format.

XmlConvert.ToDateTime(System.String, System.String) Method

```
[ILASM]
.method public hidebysig static valuetype System.DateTime
ToDateTime(string s, string format)

[C#]
public static DateTime ToDateTime(string s, string format)
```

Summary

Converts a **System.String** to a **System.DateTime** equivalent.

Parameters

Parameter	Description
<i>s</i>	The System.String to convert.
<i>format</i>	A System.String specifying the format used to validate <i>s</i> .

Return Value

The **System.DateTime** equivalent of *s*.

Description

This method calls **System.DateTime.ParseExact**(*s*, *format*, **System.Globalization.DateTimeFormatInfo.InvariantInfo**, **System.Globalization.DateTimeStyles.AllowLeadingWhite**|**System.Globalization.DateTimeStyles.AllowTrailingWhite**).

Valid formats include "yyyy-MM-ddTHH:mm:sszzzzzz" and its subsets.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.FormatException	<i>s</i> or <i>format</i> is System.String.Empty .
	-or- <i>s</i> does not contain a date and time that

1
2
3

	corresponds to <i>format</i> .
--	--------------------------------

XmlConvert.ToDateTime(System.String, System.String[]) Method

```
[ILASM]
.method public hidebysig static valuetype System.DateTime
ToDateTime(string s, class System.String[] formats)

[C#]
public static DateTime ToDateTime(string s, string[]
formats)
```

Summary

Converts a **System.String** to a **System.DateTime** equivalent.

Parameters

Parameter	Description
<i>s</i>	The System.String to convert.
<i>formats</i>	A System.String array specifying formats used to validate <i>s</i> .

Return Value

The **System.DateTime** equivalent of *s*.

Description

This method calls **System.DateTime.ParseExact**(*s*, *formats*, **System.Globalization.DateTimeFormatInfo.InvariantInfo**, **System.Globalization.DateTimeStyles.AllowLeadingWhite**|**System.Globalization.DateTimeStyles.AllowTrailingWhite**).

This method allows *s* to be validated against multiple formats.

Valid formats include "yyyy-MM-ddTHH:mm:sszzzzzz" and its subsets.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>s</i> is a null reference.
System.FormatException	<i>s</i> or an element of <i>formats</i> is System.String.Empty .

-or-

s does not contain a date and time that corresponds to any of the elements of *formats*.

Example

The following example converts a **System.String** to a **System.DateTime** and writes the result to the console.

[C#]

```
using System;
using System.Xml;

public class App {

    public static void Main() {

        String someDate = "1966-09-19T03:45:11Z";
        String[] datetimeFormats = new String[]
        { "HH:mm:ss", "yyyy-MM-ddTHH:mm:ssZ" };
        DateTime dateTime =
            XmlConvert.ToDateTime(someDate, datetimeFormats);
        Console.WriteLine("{0}", dateTime.ToString());
    }
}
```

The output is

9/18/1966 8:45:11 PM

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

XmlConvert.ToDecimal(System.String) Method

```
[ILASM]  
.method public hidebysig static decimal ToDecimal(string s)  
  
[C#]  
public static decimal ToDecimal(string s)
```

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Decimal** equivalent of s.

Description

This method calls **System.Decimal.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowDecimalPoint|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Decimal.MinValue or greater than System.Decimal.MaxValue .

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

XmlConvert.ToDouble(System.String) Method

```
[ILASM]  
.method public hidebysig static float64 ToDouble(string s)  
  
[C#]  
public static double ToDouble(string s)
```

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Double** equivalent of s.

Description

If s is "-INF", this method returns **System.Double.NegativeInfinity**.

If s is "INF", this method returns **System.Double.PositiveInfinity**.

Otherwise, this method calls **System.Double.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowDecimalPoint|System.Globalization.NumberStyles.AllowExponent|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyle s.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.

1
2
3

System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Double.MinValue or greater than System.Double.MaxValue .

XmlConvert.ToInt16(System.String)

Method

```
[ILASM]  
.method public hidebysig static int16 ToInt16(string s)  
  
[C#]  
public static short ToInt16(string s)
```

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Int16** equivalent of s.

Description

This method calls **System.Int16.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Int16.MinValue or greater than System.Int16.MaxValue .

XmlConvert.ToInt32(System.String)

Method

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

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Int32** equivalent of s.

Description

This method calls **System.Int32.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Int32.MinValue or greater than System.Int32.MaxValue .

XmlConvert.ToInt64(System.String)

Method

```
[ILASM]
.method public hidebysig static int64 ToInt64(string s)

[C#]
public static long ToInt64(string s)
```

Summary

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

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Int64** equivalent of s.

Description

This method calls **System.Int64.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Int64.MinValue or greater than System.Int64.MaxValue .

XmlConvert.ToSByte(System.String)

Method

```
[ILASM]
.method public hidebysig static int8 ToSByte(string s)

[C#]
public static sbyte ToSByte(string s)
```

Summary

Converts a **System.String** to a **System.SByte** equivalent.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.SByte** equivalent of s.

Description

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

This method calls **System.SByte.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than

1
2
3

	System.SByte.MinValue or greater than System.SByte.MaxValue .
--	--

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

XmlConvert.ToSingle(System.String) Method

```
[ILASM]  
.method public hidebysig static float32 ToSingle(string s)  
  
[C#]  
public static float ToSingle(string s)
```

Summary

Converts a **System.String** to a **System.Single** equivalent.

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.Single** equivalent of s.

Description

If s is "-INF", this method returns **System.Single.NegativeInfinity**.

If s is "INF", this method returns **System.Single.PositiveInfinity**.

Otherwise, this method calls **System.Single.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowDecimalPoint|System.Globalization.NumberStyles.AllowExponent|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.

1
2
3

System.FormatException	s is not in the correct format.
System.OverflowException	s represents a number less than System.Single.MinValue or greater than System.Single.MaxValue .

1 XmlConvert.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 to convert.

12

13

14

Return Value

15 The **System.String** "true" or the **System.String** "false".

16

1 XmlConvert.ToString(System.Char)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
<i>value</i>	The System.Char to convert.

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Description

18 This method calls *value*.ToString(**null**).

19

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

XmlConvert.ToString(System.Decimal) Method

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

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

Summary

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

Parameters

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

Return Value

The **System.String** representation of *value*.

Description

This method calls *value*.ToString(**null**,
System.Globalization.NumberFormatInfo.InvariantInfo).

1 XmlConvert.ToString(System.SByte)

2 Method

3 [ILASM]
4 .method public hidebysig static string ToString(int8 value)
5 [C#]
6 public static string ToString(sbyte value)

7 Summary

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

9 Type Attributes:

- 10
 - CLSCompliantAttribute(false)

11 Parameters

12
13

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

14
15
16

Return Value

17 The **System.String** representation of *value*.

18 Description

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

21
22 This method calls *value*.ToString(**null**,
23 **System.Globalization.NumberFormatInfo.InvariantInfo**).

24

1 XmlConvert.ToString(System.Int16)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
<i>value</i>	The System.Int16 to convert.

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Description

18 This method calls *value*.ToString(**null**,
19 **System.Globalization.NumberFormatInfo.InvariantInfo**).

20

1 XmlConvert.ToString(System.Int32)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
<i>value</i>	The System.Int32 to convert.

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Description

18 This method calls *value*.ToString(**null**,
19 **System.Globalization.NumberFormatInfo.InvariantInfo**).

20

1 XmlConvert.ToString(System.Int64)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
<i>value</i>	The System.Int64 to convert.

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Description

18 This method calls *value.ToString(null,*
19 **System.Globalization.NumberFormatInfo.InvariantInfo**).
20

XmlConvert.ToString(System.Byte)

Method

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

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

Summary

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

Parameters

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

Return Value

The **System.String** representation of *value*.

Description

This method calls *value*.ToString(**null**, **System.Globalization.NumberFormatInfo.InvariantInfo**).

1 XmlConvert.ToString(System.UInt16)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static string ToString(unsigned  
5 int16 value)  
  
6 [C#]  
7 public static string ToString(ushort value)
```

8 Summary

9 Converts a **System.UInt16** to a **System.String**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

Parameter	Description
<i>value</i>	The System.UInt16 to convert.

15
16
17

Return Value

18 The **System.String** representation of *value*.

19 Description

20 This member is not CLS-compliant. For a CLS-compliant alternative,
21 use **System.Xml.XmlConvert.ToString(System.Int32)**.

22
23 This method calls *value*.ToString(**null**,
24 **System.Globalization.NumberFormatInfo.InvariantInfo**).

25

XmlConvert.ToString(System.UInt32)

Method

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

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

Summary

Converts a **System.UInt32** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The System.UInt32 to convert.

Return Value

The **System.String** representation of *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Xml.XmlConvert.ToString(System.Int64)**.

This method calls *value.ToString(null, System.Globalization.NumberFormatInfo.InvariantInfo)*.

XmlConvert.ToString(System.UInt64)

Method

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

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

Summary

Converts a **System.UInt64** to a **System.String**.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
<i>value</i>	The System.UInt64 to convert.

Return Value

The **System.String** representation of *value*.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Xml.XmlConvert.ToString(System.Decimal)**.

This method calls *value.ToString(null, System.Globalization.NumberFormatInfo.InvariantInfo)*.

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

XmlConvert.ToString(System.Single) Method

```
[ILASM]  
.method public hidebysig static string ToString(float32  
value)  
  
[C#]  
public static string ToString(float value)
```

Summary

Converts a **System.Single** to a **System.String**.

Parameters

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

Return Value

The **System.String** representation of *value*.

Description

If *value* is **System.Double.NegativeInfinity**, this method returns "-INF".

If *value* is **System.Double.PositiveInfinity**, this method returns "INF".

Otherwise, this method calls *value*.ToString("R", **System.Globalization.NumberFormatInfo.InvariantInfo**).

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

XmlConvert.ToString(System.Double) Method

```
[ILASM]  
.method public hidebysig static string ToString(float64  
value)  
  
[C#]  
public static string ToString(double value)
```

Summary

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

Parameters

Parameter	Description
<i>value</i>	The System.Char to convert.

Return Value

The **System.String** representation of *value*.

Description

If *value* is **System.Double.NegativeInfinity**, this method returns "-INF".

If *value* is **System.Double.PositiveInfinity**, this method returns "INF".

Otherwise, this method calls *value*.ToString("R", **System.Globalization.NumberFormatInfo.InvariantInfo**).

1 XmlConvert.ToString(System.TimeSpan)

2 Method

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

8 Summary

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

10 Parameters

11
12

Parameter	Description
<i>value</i>	The System.TimeSpan to convert.

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Example

18

19 The following example converts a **System.TimeSpan** to a
20 **System.String** and writes the result to the console.

21
22

```
23 using System;
24 using System.Xml;
25
26 public class App {
27
28     public static void Main() {
29
30         TimeSpan timeSpan = new TimeSpan(3, 11, 59, 6, 128);
31         Console.WriteLine("{0}",
32             XmlConvert.ToString(timeSpan));
33     }
34 }
```

1 The output is
2
3 P3DT11H59M6.128S

4

1 XmlConvert.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** to a **System.String**.

10 Parameters

11
12

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

13
14
15

Return Value

16 The **System.String** representation of *value*.

17 Description

18 This method calls **System.Xml.XmlConvert.ToString**(*value*, "yyyy-
19 MM-ddTHH:mm:ss.fffffffzzzzz").

20

XmlConvert.ToString(System.DateTime, System.String) Method

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

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

Summary

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

Parameters

Parameter	Description
<i>value</i>	The System.DateTime to convert.
<i>format</i>	A System.String specifying the format to apply to <i>value</i> . Valid formats include "yyyy-MM-ddTHH:mm:sszzzzzz" and its subsets.

Return Value

The **System.String** representation of *value* in the specified format..

Description

This method calls *value.ToString(format, System.Globalization.DateTimeFormatInfo.InvariantInfo)*.

XmlConvert.ToTimeSpan(System.String)

Method

```
[ILASM]
.method public hidebysig static valuetype System.TimeSpan
ToTimeSpan(string s)

[C#]
public static TimeSpan ToTimeSpan(string s)
```

Summary

Converts a **System.String** to a **System.TimeSpan** equivalent.

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.TimeSpan** equivalent of s.

Exceptions

Exception	Condition
System.FormatException	s is not in the correct format to represent a System.TimeSpan value.

XmlConvert.ToUInt16(System.String)

Method

```
[ILASM]
.method public hidebysig static unsigned int16
ToUInt16(string s)

[C#]
public static ushort ToUInt16(string s)
```

Summary

Converts a **System.String** to a **System.UInt16** equivalent.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.UInt16** equivalent of s.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Xml.XmlConvert.ToInt32(System.String)**.

This method calls **System.UInt16.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.

1
2
3

System.OverflowException

s represents a number less than
System.UInt16.MinValue or greater than
System.UInt16.MaxValue.

XmlConvert.ToUInt32(System.String)

Method

```
[ILASM]
.method public hidebysig static unsigned int32
ToUInt32(string s)

[C#]
public static uint ToUInt32(string s)
```

Summary

Converts a **System.String** to a **System.UInt32** equivalent.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.UInt32** equivalent of s.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Xml.XmlConvert.ToInt64(System.String)**.

This method calls **System.UInt32.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.

1
2
3

System.OverflowException

s represents a number less than
System.UInt32.MinValue or greater than
System.UInt32.MaxValue.

XmlConvert.ToUInt64(System.String)

Method

```
[ILASM]
.method public hidebysig static unsigned int64
ToUInt64(string s)

[C#]
public static ulong ToUInt64(string s)
```

Summary

Converts a **System.String** to a **System.UInt64** equivalent.

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
s	The System.String to convert.

Return Value

The **System.UInt64** equivalent of s.

Description

This member is not CLS-compliant. For a CLS-compliant alternative, use **System.Xml.XmlConvert.ToDecimal(System.String)**.

This method calls **System.UInt64.Parse(s, System.Globalization.NumberStyles.AllowLeadingSign|System.Globalization.NumberStyles.AllowLeadingWhite|System.Globalization.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo)**.

Exceptions

Exception	Condition
System.ArgumentNullException	s is a null reference.
System.FormatException	s is not in the correct format.

1
2
3

System.OverflowException

s represents a number less than
System.UInt64.MinValue or greater than
System.UInt64.MaxValue.

XmlConvert.VerifyName(System.String)

Method

```
[ILASM]
.method public hidebysig static string VerifyName(string
name)

[C#]
public static string VerifyName(string name)
```

Summary

Verifies that the name is a valid name as defined in the W3C Extended Markup Language recommendation (REC-xml-names-19990114).

Parameters

Parameter	Description
<i>name</i>	A System.String specifying the name to verify.

Return Value

The **System.String** *name*, if it is a valid XML name.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>name</i> is null or System.String.Empty .
System.Xml.XmlException	<i>name</i> is not a valid XML name.

1 XmlConvert.VerifyNCName(System.String 2) Method

```
3 [ILASM]  
4 .method public hidebysig static string VerifyNCName(string  
5 name)  
  
6 [C#]  
7 public static string VerifyNCName(string name)
```

8 Summary

9 Verifies that the name is a valid qualified name as defined in the W3C
10 Extended Markup Language recommendation (REC-xml-names-
11 19990114).

12 Parameters

Parameter	Description
<i>name</i>	A System.String specifying the name to verify.

16 Return Value

18 The **System.String** *name*, if it is a valid XML qualified name.

19 Description

20 If *name* contains a colon, **System.Xml.XmlException** is thrown.

21 Exceptions

Exception	Condition
System.ArgumentNullException	<i>name</i> is null or System.String.Empty .
System.Xml.XmlException	<i>name</i> is not a valid XML qualified name.