

# 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:* 2.0.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](http://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

1 `System.String` to a CLI data type and vice-versa. Locale settings are not taken into  
2 account during data conversion.

3  
4 `System.Xml.XmlConvert` also provides methods that convert between XML Schema  
5 Definition (XSD) data types (see <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes>) and their corresponding common language infrastructure (CLI) data types.  
6 The following table shows the XSD data types and their corresponding CLI data types.  
7

XSD data type	CLI data type
hexBinary	A <code>System.Byte</code> array
base64Binary	A <code>System.Byte</code> array
Boolean	<code>System.Boolean</code>
Byte	<code>System.SByte</code>
normalizedString	<code>System.String</code>
Date	<code>System.DateTime</code>
duration	<code>System.TimeSpan</code>
dateTime	<code>System.DateTime</code>
decimal	<code>System.Decimal</code>
Double	<code>System.Double</code>
ENTITIES	A <code>System.String</code> array
ENTITY	<code>System.String</code>
Float	<code>System.Single</code>
gMonthDay	<code>System.DateTime</code>
gDay	<code>System.DateTime</code>
gYear	<code>System.DateTime</code>
gYearMonth	<code>System.DateTime</code>
ID	<code>System.String</code>

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 public rtspecialname specialname instance void .ctor()  
  
4 [C#]  
5 public XmlConvert()
```

## 6 Summary

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

8

# XmlConvert.DecodeName(System.String)

## Method

```
[ILAsm]  
.method public hidebysig static string DecodeName(string name)  
  
[C#]  
public static string DecodeName(string name)
```

### Summary

Decodes a name.

### Parameters

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

### Return Value

A `System.String` containing the decoded name.

### Description

Names are decoded using the following rules:

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

[*Note:* This method does the reverse of the `System.Xml.XmlConvert.EncodeName`, `System.Xml.XmlConvert.EncodeLocalName`, and `System.Xml.XmlConvert.EncodeNmToken` methods.

]

## 1   **Example**

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

5  
6       [C#]

```
7   using System;
8   using System.Xml;
9
10  public class App {
11
12       public static void Main() {
13
14           Console.WriteLine( "{0}: {1}: {2}",
15                              // _x0069_ decodes to i
16                              XmlConvert.DecodeName("Order #1_x0069_"),
17
18                              // missing beginning _
19                              XmlConvert.DecodeName("Order #1x0069_"),
20
21                              // short form
22                              XmlConvert.DecodeName("Order #1_x69_") );
23       }
24  }
```

25  
26   The output is

27  
28   Order #1i: Order #1x0069\_: Order #1\_x69\_

# XmlConvert.EncodeLocalName(System.String) Method

```
[ILAsm]  
.method public hidebysig static string EncodeLocalName(string name)  
  
[C#]  
public static string EncodeLocalName(string name)
```

## Summary

Converts a name to a valid XML local name.

## Parameters

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

## Return Value

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

## Description

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

## Example

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

```
[C#]  
  
using System;  
using System.Xml;  
  
public class App {
```

```

1
2     public static void Main() {
3
4         Console.WriteLine( "LocalName  {0}",
5             XmlConvert.EncodeLocalName( "7:+" ) );
6         Console.WriteLine( "Name  {0}",
7             XmlConvert.EncodeName( "7:+" ) );
8         Console.WriteLine( "NmToken  {0}",
9             XmlConvert.EncodeNmToken( "7:+" ) );
10    }
11 }

```

The output is

```

13
14 LocalName _x0037__x003A__x002B_
15
16 Name _x0037_:__x002B_
17
18 NmToken 7: __x002B_

```

19

# 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 <code>System.String</code> 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 might still be invalid according to the W3C Namespace Specification ([www.w3.org/TR/REC-xml-names](http://www.w3.org/TR/REC-xml-names)). To guarantee it is a

1        valid namespace qualified name use the `System.Xml.XmlConvert.EncodeLocalName`  
2        method for the prefix and local name parts and join the result with a colon.

3        **Example**

4

5        See the `System.Xml.XmlConvert.EncodeLocalName` method for an example comparing  
6        the `System.Xml.XmlConvert.EncodeLocalName`, `System.Xml.XmlConvert.EncodeName`,  
7        and `System.Xml.XmlConvert.EncodeNmToken` methods.

8

# XmlConvert.EncodeNmToken(System.String)

## Method

```
[ILAsm]  
.method public hidebysig static string EncodeNmToken(string name)  
  
[C#]  
public static string EncodeNmToken(string name)
```

### Summary

Converts a name to a valid XML name token.

### Parameters

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

### Return Value

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

### Example

See the System.Xml.XmlConvert.EncodeLocalName method for an example comparing the System.Xml.XmlConvert.EncodeLocalName, System.Xml.XmlConvert.EncodeName, and System.Xml.XmlConvert.EncodeNmToken methods.

# 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.
--	-----------------------

1

2

3

# 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.

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

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

## 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
<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s or an element of <i>formats</i> is <code>System.String.Empty</code> .  -or-  s does not contain a date and time that corresponds to any of the elements of <i>formats</i> .

## 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

# 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
s	The System.String to convert.
format	A System.String specifying the format used to validate s.

## 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:sszzzzz" and its subsets.

## Exceptions

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

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

1  
2  
3

# XmlConvert.ToDateTime(System.String)

## Method

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

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

### Summary

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

### Parameters

Parameter	Description
s	The System.String to convert.

### Return Value

The System.DateTime equivalent of s.

### Description

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

- "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"

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

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

1  
2     "---ddZ"  
3  
4     "---ddzzzzzz"  
5  
6     "--MM--"  
7  
8     "--MM--Z"  
9  
10    "--MM--zzzzzz"

11   **Exceptions**  
12  
13

Exception	Condition
<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is <code>System.String.Empty</code> or is not in the correct format.

14  
15  
16

**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.

<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.Decimal.MinValue or greater than System.Decimal.MaxValue.

1  
2  
3

**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.NumberStyles.AllowTrailingWhite, System.Globalization.NumberFormatInfo.InvariantInfo).

### Exceptions

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

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.Double.MinValue or greater than System.Double.MaxValue.

1  
2  
3

# 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.
--	------------------------

1

2

3

# 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.
--	------------------------

1

2

3

# 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.
--	------------------------

1

2

3

# 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
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.SByte.MinValue or greater than System.SByte.MaxValue.

1  
2  
3

**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
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.Single.MinValue or greater than System.Single.MaxValue.

1  
2  
3

# XmlConvert.ToString(System.Int64) Method

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

## Summary

Converts a System.Int64 to a System.String.

## Parameters

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

## Return Value

The System.String representation of *value*.

## Description

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

# XmlConvert.ToString(System.Int32) Method

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

## Summary

Converts a System.Int32 to a System.String.

## Parameters

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

## Return Value

The System.String representation of *value*.

## Description

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

# XmlConvert.ToString(System.Int16) Method

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

## Summary

Converts a System.Int16 to a System.String.

## Parameters

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

## Return Value

The System.String representation of *value*.

## Description

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

# 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 ).

# XmlConvert.ToString(System.UInt16) Method

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

## Summary

Converts a System.UInt16 to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.UInt16 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.Int32).

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

# 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).

# XmlConvert.ToString(System.TimeSpan)

## Method

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

### Summary

Converts a System.TimeSpan to a System.String.

### Parameters

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

### Return Value

The System.String representation of *value*.

### Example

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

[C#]

```
using System;  
using System.Xml;  
  
public class App {  
    public static void Main() {  
        TimeSpan timeSpan = new TimeSpan(3, 11, 59, 6, 128);  
        Console.WriteLine( "{0}",  
            XmlConvert.ToString(timeSpan) );  
    }  
}
```

1 The output is  
2  
3 P3DT11H59M6.128S  
4

# XmlConvert.ToString(System.DateTime)

## Method

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

### Summary

Converts a System.DateTime to a System.String.

### Parameters

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

### Return Value

The System.String representation of *value*.

### Description

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

# 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:sszzzzz" 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.ToString(System.SByte) Method

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

## Summary

Converts a System.SByte to a System.String.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

Parameter	Description
<i>value</i>	The System.SByte 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.Int16).

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.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 ).

# XmlConvert.ToString(System.Char) Method

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

## Summary

Converts a System.Char to a System.String.

## Parameters

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

## Return Value

The System.String representation of *value*.

## Description

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

# XmlConvert.ToString(System.Boolean)

## Method

```
[ILAsm]  
.method public hidebysig static string ToString(bool value)  
  
[C#]  
public static string ToString(bool value)
```

### Summary

Converts a System.Boolean to a System.String.

### Parameters

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

### Return Value

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

# 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.ToInt16(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
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.UInt16.MinValue or greater than System.UInt16.MaxValue.

1  
2  
3

# XmlConvert.ToInt32(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
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.UInt32.MinValue or greater than System.UInt32.MaxValue.

1  
2  
3

# 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
-----------	-----------

<b>System.ArgumentNullException</b>	s is a null reference.
<b>System.FormatException</b>	s is not in the correct format.
<b>System.OverflowException</b>	s represents a number less than System.UInt64.MinValue or greater than System.UInt64.MaxValue.

1  
2  
3

# 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.

# XmlConvert.VerifyNCName(System.String)

## Method

```
[ILAsm]  
.method public hidebysig static string VerifyNCName(string name)  
  
[C#]  
public static string VerifyNCName(string name)
```

### Summary

Verifies that the name is a valid qualified 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 qualified name.

### Description

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

### 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.