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 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. 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`

XmlConvert() Constructor

[ILAsm]
```
public rtspecialname specialname instance void .ctor()
```

[C#]
```
public XmlConvert()
```

Summary

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

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
name	A `System.String` 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.

]

Example

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

[C#]

```
using System;
using System.Xml;

public class App {

  public static void Main() {

    Console.WriteLine( "{0}: {1}: {2}",
      // _x0069_ decodes to i
      XmlConvert.DecodeName("Order #1_x0069_"),

      // missing beginning _
      XmlConvert.DecodeName("Order #1x0069_"),

      // short form
      XmlConvert.DecodeName("Order #1_x69_") );
  }
}
```

The output is

```
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
name	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 {

  public static void Main() {
```

```
  Console.WriteLine( "LocalName  {0}",
   XmlConvert.EncodeLocalName("7:+") );
  Console.WriteLine( "Name  {0}",
   XmlConvert.EncodeName("7:+") );
  Console.WriteLine( "NmToken  {0}",
   XmlConvert.EncodeNmToken("7:+") );
 }
}
```

The output is

```
LocalName _x0037__x003A__x002B_

Name _x0037_:_x002B_

NmToken 7:_x002B_
```

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
name	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 might still be invalid according to the W3C Namespace Specification (www.w3.org/TR/REC-xml-names). To guarantee it is a valid namespace qualified name use the `System.Xml.XmlConvert.EncodeLocalName` method for the prefix and local name parts and join the result with a colon.

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.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
name	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`.

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
s	The `System.String` to convert.
formats	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
System.ArgumentNullException	*s* is a null reference.
System.FormatException	*s* or an element of *formats* 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
```

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:sszzzzzz" and its subsets.

Exceptions

Exception	Condition
System.ArgumentNullException	*s* is a null reference.
System.FormatException	*s* or *format* is `System.String.Empty`. -or- *s* does not contain a date and time that corresponds to *format*.

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"

"yyyy-MM-ddTHH:mm:ssZ"

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

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

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

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

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

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

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

"yyyy-MM-ddTHH:mm:sszzzzzz"

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

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

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

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

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

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

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

"HH:mm:ss"

"HH:mm:ss.f"

"HH:mm:ss.ff"

"HH:mm:ss.fff"

"HH:mm:ss.ffff"

"HH:mm:ss.fffff"

"HH:mm:ss.ffffff"

"HH:mm:ss.fffffff"

"HH:mm:ssZ"

"HH:mm:ss.fZ"

"HH:mm:ss.ffZ"

"HH:mm:ss.fffZ"

"HH:mm:ss.ffffZ"

"HH:mm:ss.fffffZ"

"HH:mm:ss.ffffffZ"

"HH:mm:ss.fffffffZ"

"HH:mm:sszzzzzz"

"HH:mm:ss.fzzzzzz"

"HH:mm:ss.ffzzzzzz"

"HH:mm:ss.fffzzzzzz"

"HH:mm:ss.ffffzzzzzz"

"HH:mm:ss.fffffzzzzzz"

"HH:mm:ss.ffffffzzzzzz"

"HH:mm:ss.fffffffzzzzzz"

"yyyy-MM-dd"

"yyyy-MM-ddZ"

"yyyy-MM-ddzzzzzz"

"yyyy-MM"

"yyyy-MMZ"

"yyyy-MMzzzzzz"

"yyyy"

"yyyyZ"

"yyyyzzzzzz"

"--MM-dd"

"--MM-ddZ"

"--MM-ddzzzzzz"

"---dd"

"---ddZ"

"---ddzzzzzz"

"--MM--"

"--MM--Z"

"--MM--zzzzzz"

Exceptions

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

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.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.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

	`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.
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`.

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
value	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**
value	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
value	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
value	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
value	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
value	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
value	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
value	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
value	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
value	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) );
  }
}
```

The output is

```
P3DT11H59M6.128S
```

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
value	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.fffffffzzzzzz").

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
value	The `System.DateTime` to convert.
format	A `System.String` specifying the format to apply to *value*. 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.ToString(System.SByte) Method

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

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

Summary

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

Type Attributes:

- CLSCompliantAttribute(false)

Parameters

Parameter	Description
value	The `System.SByte` 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**
value	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
value	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
value	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.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.
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.
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.
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
name	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	*name* is `null` or `System.String.Empty`.
System.Xml.XmlException	*name* 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
name	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	*name* is `null` or `System.String.Empty`.
System.Xml.XmlException	*name* is not a valid XML qualified name.