System.Xml.XmlTextWriter Class

```
[ILASM]
.class public XmlTextWriter extends System.Xml.XmlWriter

[C#]
public class XmlTextWriter: XmlWriter
```

Assembly Info:

- *Name:* System.Xml
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Summary

Represents a writer that provides a fast, non-cached, forward-only way of generating streams or files containing XML data that conforms to the W3C Extensible Markup Language (XML) 1.0 and the Namespaces in XML recommendations.

Inherits From: System.Xml.XmlWriter

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

This class maintains a namespace stack corresponding to all the namespaces defined in the current element stack. Namespaces can be declared manually to override the current namespace declaration. Prefixes can be specified to associate with a namespace. If there are multiple namespace declarations mapping different prefixes to the same namespace URI, this class walks the stack of namespace declarations backwards and picks the closest one.

If namespace conflicts occur inside an element, this class resolves the conflict by generating alternate prefixes. The generated prefixes are named `ni`, where `n` is the literal character 'n' and `i` is a number beginning at one. The number is reset to one for each element. See the example section for a demonstration of this behavior.

Attributes which are associated with a namespace URI must have a prefix (default namespaces do not apply to attributes). This conforms

to section 5.2 of the W3C Namespaces in XML recommendation. If an attribute references a namespace URI, but does not specify a prefix, the writer generates a prefix for the attribute.

When writing an empty element, an additional space is added between tag name and the closing tag, for example `<item />`. This provides compatibility with older browsers.

When a **System.String** is used as method parameter, **null** and **System.String.Empty** are equivalent. **System.String.Empty** follows the W3C rules.

This class implements the **System.Xml.XmlWriter** class.

Example

The following example demonstrates how this class resolves namespace conflicts inside an element. In the example, the writer writes an element that contains two attributes. The element and both attributes have the same prefix but different namespaces. The resulting XML fragment is written to the console.

[C#]

```
using System;
using System.Xml;

public class WriteFragment
{
  public static void Main()
  {
    XmlTextWriter xWriter = new XmlTextWriter(Console.Out);
    xWriter.WriteStartElement("prefix", "Element1",
"namespace");
    xWriter.WriteStartAttribute("prefix", "Attr1",
"namespace1");
    xWriter.WriteString("value1");
    xWriter.WriteStartAttribute("prefix", "Attr2",
"namespace2");
    xWriter.WriteString("value2");
    xWriter.Close();
  }
}
```

The output is

```
<prefix:Element1 n1:Attr1="value1" n2:Attr2="value2"
```

```
        xmlns:n2="namespace2" xmlns:n1="namespace1"
        xnlns:prefix="namespace" />

```

XmlTextWriter(System.IO.Stream, System.Text.Encoding) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.IO.Stream w, class System.Text.Encoding encoding)

[C#]
public XmlTextWriter(Stream w, Encoding encoding)
```

Summary

Constructs and initializes a new instance of the **System.Xml.XmlTextWriter** class using the specified output stream.

Parameters

Parameter	Description
w	The **System.IO.Stream** to write to.
encoding	The **System.Text.Encoding** to generate, or **null**.

Description

If *encoding* is **null**, the stream is written as UTF-8 and the encoding attribute is omitted from the processing instruction.

The following properties are initialized to the specified values:

System.Xml.XmlTextWriter.Formatting to **System.Xml.Formatting.None**.

System.Xml.XmlTextWriter.Indentation to 2.

System.Xml.XmlTextWriter.IndentChar to the space character.

System.Xml.XmlTextWriter.Namespaces to **true**.

System.Xml.XmlTextWriter.QuoteChar to the double quote character.

System.Xml.XmlTextWriter.WriteState to **System.Xml.WriteState.Start**.

Exceptions

Exception	Condition

System.ArgumentException	*w* cannot be written to. -or- The encoding is not supported.
System.ArgumentNullException	*w* is **null**.

1
2
3

XmlTextWriter(System.String, System.Text.Encoding) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(string
filename, class System.Text.Encoding encoding)

[C#]
public XmlTextWriter(string filename, Encoding encoding)
```

Summary

Constructs and initializes a new instance of the **System.Xml.XmlTextWriter** class using the specified file.

Parameters

Parameter	Description
filename	A **System.String** specifying the path and name of the file to write to.
encoding	The **System.Text.Encoding** to generate, or **null**.

Description

If *filename* exists, it is truncated and overwritten with the new content.

If *encoding* is **null**, the file is written as UTF-8 and the encoding attribute is omitted from the processing instruction.

The following properties are initialized to the specified values:

System.Xml.XmlTextWriter.Formatting to **System.Xml.Formatting.None**.

System.Xml.XmlTextWriter.Indentation to 2.

System.Xml.XmlTextWriter.IndentChar to the space character.

System.Xml.XmlTextWriter.Namespaces to **true**.

System.Xml.XmlTextWriter.QuoteChar to the double quote character.

System.Xml.XmlTextWriter.WriteState to **System.Xml.WriteState.Start**.

Exceptions

Exception	Condition
System.ArgumentException	*filename* is **System.String.Empty**, contains only white space, or contains one or more implementation-defined invalid characters. -or- The encoding is not supported.
System.ArgumentNullException	*filename* is **null**.
System.IO.DirectoryNotFoundException	The directory path specified in *filename* does not exist.
System.IO.IOException	*filename* includes an invalid syntax for the path or file name.
System.IO.PathTooLongException	The specified path, file name, or both exceeds the system-defined maximum length.
System.Security.SecurityException	The caller does not have the required permissions.
System.UnauthorizedAccessException	Write access is not permitted by the operating system for the path specified in *filename*.

Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission to write to files. See **System.Security.Permissions.FileIOPermissionAccess Write**.

XmlTextWriter(System.IO.TextWriter) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.IO.TextWriter w)

[C#]
public XmlTextWriter(TextWriter w)
```

Summary

Constructs and initializes a new instance of the **System.Xml.XmlTextWriter** class.

Parameters

Parameter	Description
w	The **System.IO.TextWriter** to write to, initialized to the correct encoding.

Description

The following properties are initialized to the specified values:

System.Xml.XmlTextWriter.Formatting to **System.Xml.Formatting.None**.

System.Xml.XmlTextWriter.Indentation to 2.

System.Xml.XmlTextWriter.IndentChar to the space character.

System.Xml.XmlTextWriter.Namespaces to **true**.

System.Xml.XmlTextWriter.QuoteChar to the double quote character.

System.Xml.XmlTextWriter.WriteState to **System.Xml.WriteState.Start**.

[*Note:* If a specific encoding is necessary, set the encoding using the constructor of *w* before instantiating the writer.]

XmlTextWriter.Close() Method

```
[ILASM]
.method public hidebysig virtual void Close()

[C#]
public override void Close()
```

Summary

Closes the writer.

Description

This method closes all elements and attributes created by the **System.Xml.XmlTextWriter.WriteStartElement** and **System.Xml.XmlTextWriter.WriteStartAttribute** methods, respectively, that are open when the **System.Xml.XmlTextWriter.Close** method is called.

This method calls the **System.Xml.XmlTextWriter.Flush** method to flush the underlying buffered stream and then closes the stream.

This method sets the **System.Xml.XmlTextWriter.WriteState** to **System.Xml.WriteState.Closed**.

XmlTextWriter.Flush() Method

```
[ILASM]
.method public hidebysig virtual void Flush()

[C#]
public override void Flush()
```

Summary

Clears all buffers and causes any buffered data to be written to the underlying stream.

Description

[*Note:* This method overrides **System.Xml.XmlWriter.Flush**.]

XmlTextWriter.LookupPrefix(System.String) Method

```
[ILASM]
.method public hidebysig virtual string LookupPrefix(string
ns)

[C#]
public override string LookupPrefix(string ns)
```

Summary

Returns the prefix defined in the current namespace scope for the specified namespace URI.

Parameters

Parameter	Description
ns	A **System.String** specifying the namespace URI.

Return Value

A **System.String** containing the corresponding prefix, or **System.String.Empty** if the prefix is not found and *ns* is the default namespace, or **null** if no matching namespace URI is found in the current scope.

Description

[*Note:* This method overrides **System.Xml.XmlWriter.LookupPrefix**.]

Exceptions

Exception	Condition
System.ArgumentException	*ns* is **null** or **System.String.Empty**.

XmlTextWriter.WriteBase64(System.Byte[], System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void WriteBase64(class
System.Byte[] buffer, int32 index, int32 count)

[C#]
public override void WriteBase64(byte[] buffer, int index,
int count)
```

Summary

Encodes the specified binary bytes as Base64 and writes the resulting text.

Parameters

Parameter	Description
buffer	A **System.Byte** array containing the bytes to encode.
index	A **System.Int32** specifying the position within the array of the first byte to encode.
count	A **System.Int32** specifying the number of bytes to encode.

Description

[*Note:* Base64 encoding represents byte sequences in a text form comprised of the 65 US-ASCII characters (A-Z, a-z, 0-9, +, /, =) where each character encodes 6 bits of the binary data.

For more information on Base64 encoding, see RFC 2045 (http://www.ietf.org/rfc/rfc2045).

This method overrides **System.Xml.XmlWriter.WriteBase64**.]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is **null**.
System.ArgumentException	The buffer length minus *index* is less than *count*.
System.ArgumentOutOfRangeException	*index* or *count* is less than zero.
System.InvalidOperationException	The

	System.Xml.XmlTextWriter.WriteState is **System.Xml.WriteState.Closed**.

1
2
3

XmlTextWriter.WriteBinHex(System.Byte[], System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void WriteBinHex(class
System.Byte[] buffer, int32 index, int32 count)

[C#]
public override void WriteBinHex(byte[] buffer, int index,
int count)
```

Summary

Encodes the specified binary bytes as BinHex and writes the resulting text.

Parameters

Parameter	Description
buffer	A **System.Byte** array containing the bytes to encode.
index	A **System.Int32** specifying the position within the array of the first byte to encode.
count	A **System.Int32** specifying the number of bytes to encode.

Description

[*Note:* For information on BinHex encoding, see RFC 1741 (http://www.ietf.org/rfc/rfc1741).

This method overrides **System.Xml.XmlWriter.WriteBinHex**.]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is **null**.
System.ArgumentException	The buffer length minus *index* is less than *count*.
System.ArgumentOutOfRangeException	*index* or *count* is less than zero.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteCData(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteCData(string
text)

[C#]
public override void WriteCData(string text)
```

Summary

Writes out a CDATA block containing the specified text.

Parameters

Parameter	Description
text	A **System.String** specifying the text to place inside the CDATA block.

Description

This method writes <![CDATA[*text*]]>.

If *text* is **null** or **System.String.Empty**, this method writes an empty CDATA block, <![CDATA[]]>.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteCData**.]

Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteCharEntity(System.Char) Method

```
[ILASM]
.method public hidebysig virtual void
WriteCharEntity(valuetype System.Char ch)

[C#]
public override void WriteCharEntity(char ch)
```

Summary

Forces the generation of a character entity for the specified Unicode character value.

Parameters

Parameter	Description
ch	The **System.Char** for which to generate the entity.

Description

This method writes the Unicode character in hexadecimal character entity reference format.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteCharEntity**.]

Exceptions

Exception	Condition
System.ArgumentException	The character is in the surrogate pair character range, 0xd800 - 0xdfff, or the text would result in a non-well formed XML document.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteChars(System.Char[], System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void WriteChars(class
System.Char[] buffer, int32 index, int32 count)

[C#]
public override void WriteChars(char[] buffer, int index,
int count)
```

Summary

Writes text a buffer at a time.

Parameters

Parameter	Description
buffer	A **System.Char** array containing the text to write.
index	A **System.Int32** specifying the position within the array of the start of the text to write.
count	A **System.Int32** specifying the number of characters to write.

Description

[*Note:* This method can be used to write large amounts of text a buffer at a time.

An exception is thrown if surrogate pair characters would be split across multiple buffer writes. This exception must be caught in order to continue writing the next surrogate pair characters. The XML specification defines the valid ranges for surrogate pairs.

This method overrides **System.Xml.XmlWriter.WriteChars**.]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is **null**.
System.ArgumentOutOfRangeException	*index* or *count* is less than zero. - or - The buffer length minus *index* is less than

	count.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

1
2
3

XmlTextWriter.WriteComment(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteComment(string
text)

[C#]
public override void WriteComment(string text)
```

Summary

Writes out a comment containing the specified text.

Parameters

Parameter	Description
text	A **System.String** containing the text to place inside the comment.

Description

This method writes <!-- *text* -->.

If *text* is **null** or **System.String.Empty**, this method writes a comment with no content, <!---->.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteComment**.]

Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteDocType(System.String, System.String, System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteDocType(string
name, string pubid, string sysid, string subset)

[C#]
public override void WriteDocType(string name, string
pubid, string sysid, string subset)
```

Summary

Writes the document type declaration with the specified name and optional attributes.

Parameters

Parameter	Description
name	A **System.String** specifying the name of the document type.
pubid	A **System.String** specifying the public identifier, which is an alternative to the system identifier.
sysid	A **System.String** specifying the system identifier, which is the URI of the DTD (document type definition) for the document.
subset	A **System.String** specifying a URI that contains markup declarations.

Description

The optional attributes, *pubid*, *sysid*, and *subset*, are not checked for invalid characters.

[*Note:* A document type declaration is of the following form:

<!DOCTYPE *name* PUBLIC "*pubid*" "*sysid*" [*subset*]>

This method overrides **System.Xml.XmlWriter.WriteDocType**.]

Exceptions

Exception	Condition

System.ArgumentException	*name* is **null** or **System.String.Empty**. -or- The value for *name* would result in invalid XML.
System.InvalidOperationException	This method was called outside the prolog (after the root element).

1
2
3

XmlTextWriter.WriteEndAttribute() Method

```
[ILASM]
.method public hidebysig virtual void WriteEndAttribute()

[C#]
public override void WriteEndAttribute()
```

Summary

Closes the attribute started with the **System.Xml.XmlTextWriter.WriteStartAttribute** method.

Description

[*Note:* The **System.Xml.XmlTextWriter.WriteStartAttribute** and **System.Xml.XmlTextWriter.WriteEndElement** methods also will close an open attribute if one exists when they are called.

This method overrides **System.Xml.XmlWriter.WriteEndAttribute**.]

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteEndDocument() Method

```
[ILASM]
.method public hidebysig virtual void WriteEndDocument()

[C#]
public override void WriteEndDocument()
```

Summary

Closes open elements and attributes and sets the **System.Xml.XmlTextWriter.WriteState** back to the **System.Xml.WriteState.Start** state.

Description

This method closes all elements and attributes created by the **System.Xml.XmlTextWriter.WriteStartElement** and **System.Xml.XmlTextWriter.WriteStartAttribute** methods, respectively, that are open when the **System.Xml.XmlTextWriter.WriteEndDocument** method is called.

[*Note:* After calling this method, the current instance can be used to write a new XML document.

This method overrides **System.Xml.XmlWriter.WriteEndDocument**.]

Exceptions

Exception	Condition
System.ArgumentException	The current instance is in the wrong **System.Xml.WriteState**, or the document does not have a root element.

XmlTextWriter.WriteEndElement() Method

```
[ILASM]
.method public hidebysig virtual void WriteEndElement()

[C#]
public override void WriteEndElement()
```

Summary

Closes an open element and pops the corresponding namespace scope.

Description

If the open element does not contain content, it is closed as an empty element using " />"; otherwise an end element is written.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteEndElement**.]

Exceptions

Exception	Condition
System.InvalidOperationException	No element was open, or the **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteEntityRef(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteEntityRef(string
name)

[C#]
public override void WriteEntityRef(string name)
```

Summary

Writes an entity reference with the specified name.

Parameters

Parameter	Description
name	A **System.String** specifying the name of the entity reference.

Description

This method writes % *name*;.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteEntityRef**.]

Exceptions

Exception	Condition
System.ArgumentException	A **System.String** containing the text would result in a non-well formed XML document, or *name* is either **null** or **System.String.Empty**.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteFullEndElement() Method

```
[ILASM]
.method public hidebysig virtual void WriteFullEndElement()

[C#]
public override void WriteFullEndElement()
```

Summary

Closes an open element and pops the corresponding namespace scope.

Description

This method writes an end element regardless of whether there is any content in the element.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteFullEndElement**.]

Exceptions

Exception	Condition
System.InvalidOperationException	No start tag was open, or the **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteName(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteName(string
name)

[C#]
public override void WriteName(string name)
```

Summary

Writes out the specified name, ensuring it is a valid name according to the W3C XML 1.0 recommendation (http://www.w3.org/TR/1998/REC-xml-19980210#NT-Name).

Parameters

Parameter	Description
name	A **System.String** specifying the name to write.

Description

If **System.Xml.XmlTextWriter.Namespaces** is set to **true**, this method checks that *name* is also valid according to the W3C Namespaces in XML recommendation (http://www.w3.org/TR/REC-xml-names).

[*Note:* This method overrides **System.Xml.XmlWriter.WriteName**.]

Exceptions

Exception	Condition
System.ArgumentException	*name* is **null** or **System.String.Empty**; or *name* is not a valid XML Name.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteNmToken(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteNmToken(string
name)

[C#]
public override void WriteNmToken(string name)
```

Summary

Writes out the specified name, ensuring it is a valid name token (Nmtoken) according to the W3C XML 1.0 recommendation (http://www.w3.org/TR/1998/REC-xml-19980210#NT-Name).

Parameters

Parameter	Description
name	A **System.String** specifying the name to write.

Description

[*Note:* This method overrides **System.Xml.XmlWriter.WriteNmToken**.]

Exceptions

Exception	Condition
System.ArgumentException	*name* is **null** or **System.String.Empty**; or *name* is not a valid XML Nmtoken.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteProcessingInstruction(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void
WriteProcessingInstruction(string name, string text)

[C#]
public override void WriteProcessingInstruction(string
name, string text)
```

Summary

Writes out a processing instruction with the specified name and text.

Parameters

Parameter	Description
name	A **System.String** specifying the name of the processing instruction.
text	A **System.String** specifying the text to include in the processing instruction.

Description

This method writes <? *name text* ?>.

If *text* is **null** or **System.String.Empty**, this method writes a processing instruction with no text content, <? *name* ?>.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteProcessingInstruction**.]

Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document. - or - *name* is **null** or **System.String.Empty**. - or - This method is being used to create an XML

	declaration after **System.Xml.XmlTextWriter.WriteStartDocumen** has already been called.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

1
2
3

XmlTextWriter.WriteQualifiedName(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void
WriteQualifiedName(string localName, string ns)

[C#]
public override void WriteQualifiedName(string localName,
string ns)
```

Summary

Writes out the qualified name.

Parameters

Parameter	Description
localName	A **System.String** specifying the local name to write.
ns	A **System.String** specifying the namespace URI to associate with *localname*.

Description

If *ns* maps to the current default namespace, no prefix is generated.

When writing attribute values, this method generates a prefix if *ns* is not found. When writing element content, this method throws an exception if *ns* is not found.

If the current instance supports namespaces (**System.Xml.XmlTextWriter.Namespaces** is set to **true**), this method looks up the prefix that is in scope for the given namespace and checks that the name is valid according to the W3C Namespaces in XML recommendation (http://www.w3.org/TR/REC-xml-names).

[*Note:* This method overrides **System.Xml.XmlWriter.WriteQualifiedName**.]

Exceptions

Exception	Condition
System.ArgumentException	*localName* is **null** or **System.String.Empty**. -or-

	System.Xml.XmlTextWriter.Namespaces is **false**, and *ns* is neither **null** nor **System.String.Empty**.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.
System.Xml.XmlException	*localName* is not a valid XML name.

1
2
3

XmlTextWriter.WriteRaw(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteRaw(string data)

[C#]
public override void WriteRaw(string data)
```

Summary

Writes raw text from a string.

Parameters

Parameter	Description
data	A **System.String** specifying the text to write.

Description

If *data* is **null**, **System.String.Empty** is written.

This method does not encode any characters.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteRaw**.]

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteRaw(System.Char[], System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void WriteRaw(class
System.Char[] buffer, int32 index, int32 count)

[C#]
public override void WriteRaw(char[] buffer, int index, int
count)
```

Summary

Writes raw text from a character array.

Parameters

Parameter	Description
buffer	A **System.Char** array containing the text to write.
index	A **System.Int32** specifying the position within the array of the start of the text to write.
count	A **System.Int32** specifying the number of characters to write.

Description

This method does not encode any characters.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteRaw**.]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is **null**.
System.ArgumentOutOfRangeException	*index* or *count* is less than zero. - or - The buffer length minus *index* is less than *count*.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

1
2
3

XmlTextWriter.WriteStartAttribute(System.String, System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void
WriteStartAttribute(string prefix, string localName, string
ns)

[C#]
public override void WriteStartAttribute(string prefix,
string localName, string ns)
```

Summary

Writes the start of an attribute with the specified prefix and name, and associates the prefix with the specified namespace URI.

Parameters

Parameter	Description
prefix	A **System.String** specifying the namespace prefix of the attribute.
localName	A **System.String** specifying the local name of the attribute.
ns	A **System.String** specifying the namespace URI associated with the attribute.

Description

If any of the input parameters are **null** or **System.String.Empty**, the start attribute is written with that parameter missing.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteStartAttribute**.]

Exceptions

Exception	Condition
System.ArgumentException	**System.Xml.XmlTextWriter.Namespaces** is **false** for the writer, and *prefix* and *ns* are not both **null**.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

1
2
3

XmlTextWriter.WriteStartDocument(System.Boolean) Method

```
[ILASM]
.method public hidebysig virtual void
WriteStartDocument(bool standalone)

[C#]
public override void WriteStartDocument(bool standalone)
```

Summary

Writes the XML declaration with the version "1.0" and the standalone attribute.

Parameters

Parameter	Description
standalone	A **System.Boolean** where **true** indicates to write "yes" as the value for the standalone attribute, and **false** indicates to write "no".

Description

[*Note:* When **System.Xml.XmlTextWriter** is instantiated, the **System.Xml.XmlTextWriter.WriteState** is set to **System.Xml.WriteState.Start**. All the "write" methods change the **System.Xml.XmlTextWriter.WriteState** to a value other than **Start**. Thus, if this method is not the first "write" method called, a **System.InvalidOperationException** is thrown.

If **System.Xml.XmlTextWriter.WriteStartDocument** has been called and then the **System.Xml.XmlTextWriter.WriteProcessingInstruction** method is used to create another XML declaration, a **System.ArgumentException** will be thrown.

The output of this method with *standalone* equal to **true**, an encoding equal to **System.Text.Encoding.Unicode**, and using the default **System.Xml.XmlTextWriter.QuoteChar** is:

```
<?xml version="1.0" encoding="utf-16" standalone="yes"?>
```

Character encoding is set when the writer is instantiated.

This method overrides **System.Xml.XmlWriter.WriteStartDocument**.]

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is not **System.Xml.WriteState.Start**.

XmlTextWriter.WriteStartDocument() Method

```
[ILASM]
.method public hidebysig virtual void WriteStartDocument()

[C#]
public override void WriteStartDocument()
```

Summary

Writes the XML declaration with the version "1.0" and no standalone attribute.

Description

[*Note:* When **System.Xml.XmlTextWriter** is instantiated, the **System.Xml.XmlTextWriter.WriteState** is set to **System.Xml.WriteState.Start**. All the "write" methods change the **System.Xml.XmlTextWriter.WriteState** to a value other than **Start**. Thus, if this method is not the first "write" method called, a **System.InvalidOperationException** is thrown.

If **System.Xml.XmlTextWriter.WriteStartDocument** has been called and then the **System.Xml.XmlTextWriter.WriteProcessingInstruction** method is used to create another XML declaration, a **System.ArgumentException** will be thrown.

The output of this method using an encoding equal to **System.Text.Encoding.Unicode** and the default **System.Xml.XmlTextWriter.QuoteChar** is

```
<?xml version="1.0" encoding="utf-16"?>
```

Character encoding is set when the writer is instantiated.

This method overrides **System.Xml.XmlWriter.WriteStartDocument**.]

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is not **System.Xml.WriteState.Start**.

1
2
3

XmlTextWriter.WriteStartElement(System.String, System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void
WriteStartElement(string prefix, string localName, string
ns)

[C#]
public override void WriteStartElement(string prefix,
string localName, string ns)
```

Summary

Writes a start element with the specified name, and associates it with the given namespace and prefix.

Parameters

Parameter	Description
prefix	A **System.String** specifying the namespace prefix of the element.
localName	A **System.String** specifying the local name of the element.
ns	A **System.String** specifying the namespace URI to associate with the element.

Description

If *ns* is already in scope and has an associated prefix, the current instance will automatically write that prefix also.

If any of the input parameters are **null** or **System.String.Empty**, the start element is written with that parameter missing.

[*Note:* Write any attributes using the **System.Xml.XmlTextWriter.WriteStartAttribute**, **System.Xml.XmlTextWriter.WriteString**, and **System.Xml.XmlTextWriter.WriteEndAttribute** methods, then close the element using the **System.Xml.XmlTextWriter.WriteEndElement** method.

This method overrides **System.Xml.XmlWriter.WriteStartElement**.]

Exceptions

Exception	Condition
System.ArgumentException	**System.Xml.XmlTextWriter.Namespaces** is **false** for the writer, and *prefix* and *ns* are not both **null**.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

Example

This example demonstrates the **System.Xml.XmlTextWriter.WriteStartElement** method, writing the XML to the console.

[C#]

```
using System;
using System.Xml;

public class WriteXml
{
  public static void Main()
  {
    XmlTextWriter xWriter =
      new XmlTextWriter(Console.Out);
    xWriter.WriteStartDocument();
    xWriter.WriteStartElement("prefix","element",
"namespace");
    xWriter.WriteEndDocument();
  }
}
```

The output is

```
<?xml version="1.0" encoding= "someencoding"?>

<prefix:element xmlns:prefix="namespace" />
```

The value of the encoding attribute is the encoding of the output stream of the console.

1

XmlTextWriter.WriteString(System.String) Method

```
[ILASM]
.method public hidebysig virtual void WriteString(string
text)

[C#]
public override void WriteString(string text)
```

Summary

Writes the specified text.

Parameters

Parameter	Description
text	A **System.String** specifying the text to write.

Description

This method performs the following conversions before writing the text:

- The characters '&', '<', and '>' are replaced with "&", "<", and ">", respectively.
- Character values in the range 0x-0x1F (excluding the white space characters 0x9, 0x10, and 0x13) are replaced with numeric character entities ("&#0;" through "&#0x1F").
- If called in the context of an attribute value, double and single quotes are replaced with """ and "'" respectively.

If *text* is **null** or **System.String.Empty**, this method writes a text node with no data content.

[*Note:* This method overrides **System.Xml.XmlWriter.WriteString**.]

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed** and *text* is neither **null** nor **System.String.Empty**.

Example

The following example demonstrates the conversions performed by this method.

[C#]

```
using System;
using System.Xml;

public class WriteFrag {

  public static void Main() {

    XmlTextWriter xtWriter =
      new XmlTextWriter(Console.Out);
    xtWriter.WriteString("<1 & 2 = 3>");
  }
}
```

The output is

```
&lt;1 &amp; 2 = 3&gt;
```

XmlTextWriter.WriteSurrogateCharEntity(System.Char, System.Char) Method

```
[ILASM]
.method public hidebysig virtual void
WriteSurrogateCharEntity(valuetype System.Char lowChar,
valuetype System.Char highChar)

[C#]
public override void WriteSurrogateCharEntity(char lowChar,
char highChar)
```

Summary

Generates and writes the surrogate character entity for the surrogate character pair.

Parameters

Parameter	Description
lowChar	A **System.Char** containing the low surrogate. This must be a value between 0xDC00 and 0xDFFF.
highChar	A **System.Char** containing the high surrogate. This must be a value between 0xD800 and 0xDBFF.

Description

This method only applies to a writer that uses the UTF-16 encoding type.

The surrogate character entity is written in hexadecimal format. The range for surrogate characters is #x10000 to #x10FFFF. The following formula is used to generate the surrogate character entity: (*highChar* `- 0xD800) * 0x400 + (`*lowChar* `- 0xDC00) + 0x10000`.

[*Note:* For both HTML and XML, the document character set (and therefore the notation of numeric character references) is based on UCS [ISO-10646]. A single numeric character reference in a source document may therefore in some cases correspond to two 16-bit units in a string (a high surrogate and a low surrogate). These 16-bit units are referred to as a surrogate pair.

For more information regarding surrogates or characters, refer to section 3.7 of the Unicode 3.0/Unicode 2.0 standard located at http://www.unicode.org, or section 2.2 of the W3C XML 1.0 Recommendation located at http://www.w3.org/TR/REC-xml#charsets.

This method overrides **System.Xml.XmlWriter.WriteSurrogateCharEntity**.]

Exceptions

Exception	Condition
System.ArgumentException	An invalid surrogate character pair was passed.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.WriteWhitespace(System.String) Method

```
[ILASM]
.method public hidebysig virtual void
WriteWhitespace(string ws)

[C#]
public override void WriteWhitespace(string ws)
```

Summary

Writes the given white space.

Parameters

Parameter	Description
ws	A **System.String** containing the white space characters.

Description

[*Note:* This method is used to manually format a document. Use the **System.Xml.XmlTextWriter.Formatting** property to have the current instance format the output automatically.

This method overrides **System.Xml.XmlWriter.WriteWhitespace**.]

Exceptions

Exception	Condition
System.ArgumentException	*ws* is **null** or **System.String.Empty** or contains non-white space characters.
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** is **System.Xml.WriteState.Closed**.

XmlTextWriter.BaseStream Property

```
[ILASM]
.property class System.IO.Stream BaseStream { public
hidebysig specialname instance class System.IO.Stream
get_BaseStream() }

[C#]
public Stream BaseStream { get; }
```

Summary

Gets the underlying stream used by the writer.

Property Value

A **System.IO.Stream**, or **null** if the current instance does not use an underlying stream.

Description

This property is read-only.

If the current instance was constructed using a **System.IO.TextWriter** that is a subclass of the **System.IO.StreamWriter** class, this property is equivalent to the **System.IO.StreamWriter.BaseStream** property.

If the writer was constructed using a **System.IO.Stream**, this property returns the **Stream** passed to the constructor.

If the writer was constructed using a file name, this property returns the **Stream** representing the file.

XmlTextWriter.Formatting Property

```
[ILASM]
.property valuetype System.Xml.Formatting Formatting {
public hidebysig specialname instance valuetype
System.Xml.Formatting get_Formatting() public hidebysig
specialname instance void set_Formatting(valuetype
System.Xml.Formatting value) }

[C#]
public Formatting Formatting { get; set; }
```

Summary

Indicates how the output is formatted.

Property Value

One of the members of the **System.Xml.Formatting** enumeration. The default is **System.Xml.Formatting.None** (no special formatting).

Description

If this property is set to **System.Xml.Formatting.Indented**, child elements are indented using the **System.Xml.XmlTextWriter.Indentation** and **System.Xml.XmlTextWriter.IndentChar** properties. Only element content will be indented.

[*Note:* Writing any text content, including **System.String.Empty**, puts that element into mixed content mode. Child elements do not inherit this "mixed" mode status. A child element of a "mixed" element will do indenting, unless it is also contains "mixed" content. Element content (http://www.w3.org/TR/1998/REC-xml-19980210#sec-element-content) and mixed content (http://www.w3.org/TR/1998/REC-xml-19980210#sec-mixed-content) are defined according to the XML 1.0 definitions of these terms.]

XmlTextWriter.Indentation Property

```
[ILASM]
.property int32 Indentation { public hidebysig specialname
instance int32 get_Indentation() public hidebysig
specialname instance void set_Indentation(int32 value) }

[C#]
public int Indentation { get; set; }
```

Summary

Gets or sets how many indentation characters to write for each level in the hierarchy when **System.Xml.XmlTextWriter.Formatting** is set to **System.Xml.Formatting.Indented**.

Property Value

A **System.Int32** specifying the number of **System.Xml.XmlTextWriter.IndentChar** characters to use for each level. The default is 2.

Description

Indentation is performed on the following members of **System.Xml.XmlNodeType**: **DocumentType**, **Element**, **Comment**, **ProcessingInstruction**, and **CDATA**. All other node types are not affected. The **System.Xml.XmlTextWriter** class does not indent the internal DTD subset.

Exceptions

Exception	Condition
System.ArgumentException	The value to be set is less than zero.

XmlTextWriter.IndentChar Property

```
[ILASM]
.property valuetype System.Char IndentChar { public
hidebysig specialname instance valuetype System.Char
get_IndentChar() public hidebysig specialname instance void
set_IndentChar(valuetype System.Char value) }

[C#]
public char IndentChar { get; set; }
```

Summary

Gets or sets the character to use for indenting when **System.Xml.XmlTextWriter.Formatting** is set to **System.Xml.Formatting.Indented**.

Property Value

A **System.Char** specifying the character to use for indenting. The default is space (character code 0x20).

Description

[*Note:* This property can be set to any character. To ensure valid XML, set this property to a valid white space character: 0x9, 0x10, 0x13, or 0x20.]

XmlTextWriter.Namespaces Property

```
[ILASM]
.property bool Namespaces { public hidebysig specialname
instance bool get_Namespaces() public hidebysig specialname
instance void set_Namespaces(bool value) }

[C#]
public bool Namespaces { get; set; }
```

Summary

Gets or sets a value indicating whether the writer supports namespaces.

Property Value

A **System.Boolean** where **true** indicates the writer supports namespaces; otherwise, **false**. The default is **true**.

Description

This property determines whether the writer supports the XML Namespaces specification (http://www.w3.org/TR/REC-xml-names).

If an attempt is made to set this property after a write operation has occurred, a **System.InvalidOperationException** is thrown.

Exceptions

Exception	Condition
System.InvalidOperationException	The **System.Xml.XmlTextWriter.WriteState** of the current instance is not **System.Xml.WriteState.Start**.

XmlTextWriter.QuoteChar Property

```
[ILASM]
.property valuetype System.Char QuoteChar { public
hidebysig specialname instance valuetype System.Char
get_QuoteChar() public hidebysig specialname instance void
set_QuoteChar(valuetype System.Char value) }

[C#]
public char QuoteChar { get; set; }
```

Summary

Gets or sets the character used to quote the value of an attribute.

Property Value

A **System.Char** specifying the quotation mark character (" or ') used to enclose the value of an attribute. The default is the double quote.

Exceptions

Exception	Condition
System.ArgumentException	The value to be set is not the single quote or double quote character.

XmlTextWriter.WriteState Property

```
[ILASM]
.property valuetype System.Xml.WriteState WriteState {
public hidebysig virtual specialname valuetype
System.Xml.WriteState get_WriteState() }

[C#]
public override WriteState WriteState { get; }
```

Summary

Gets the write state of the writer.

Property Value

One of the members of the **System.Xml.WriteState** enumeration.

Description

This property is read-only.

[*Note:* This property overrides **System.Xml.XmlWriter.WriteState**.]

XmlTextWriter.XmlLang Property

```
[ILASM]
.property string XmlLang { public hidebysig virtual
specialname string get_XmlLang() }

[C#]
public override string XmlLang { get; }
```

Summary

Gets the language attribute, `xml:lang`, specifying the language in which the content and attribute values of the current element are written.

Property Value

A **System.String** containing the language attribute, or **null** if the language attribute is not specified for the element.

Description

This property is read-only.

[*Note:* This property overrides **System.Xml.XmlWriter.XmlLang**.]

XmlTextWriter.XmlSpace Property

```
[ILASM]
.property valuetype System.Xml.XmlSpace XmlSpace { public
hidebysig virtual specialname valuetype System.Xml.XmlSpace
get_XmlSpace() }

[C#]
public override XmlSpace XmlSpace { get; }
```

Summary

Gets the white space attribute, `xml:space`, specifying how white space is handled in the current element.

Property Value

One of the members of the **System.Xml.XmlSpace** enumeration, or **System.Xml.XmlSpace.None** if the white space attribute is not specified for the element.

Description

This property is read-only.

[*Note:* This property overrides **System.Xml.XmlWriter.XmlSpace**.]