

1 System.Xml.XmlWriter Class

```
2 [ILAsm]  
3 .class public abstract XmlWriter extends System.Object  
4 [C#]  
5 public abstract class XmlWriter
```

6 Assembly Info:

- 7 • *Name:* System.Xml
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Represents a writer that provides a non-cached, forward-only means of generating
14 streams or files containing XML data.

15 Inherits From: System.Object

16

17 **Library:** XML

18

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

21

22 Description

23 The output of this class conforms to the W3C Extensible Markup Language (XML) 1.0
24 and the Namespaces in XML recommendations.

25

26 [*Note:* This class is abstract and is implemented in the System.Xml.XmlTextWriter
27 class.

28

29]

30

1 XmlWriter() Constructor

```
2 [ILAsm]  
3 family rtspecialname specialname instance void .ctor()  
4 [C#]  
5 protected XmlWriter()
```

6 Summary

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

8

1 XmlWriter.Close() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void Close()  
4 [C#]  
5 public abstract void Close()
```

6 Summary

7 Closes the writer.

8 Behaviors

9 This method closes any remaining open elements or attributes.

10
11 This method calls the `System.Xml.XmlWriter.Flush` method to flush the underlying
12 buffered stream and then closes the stream.

13
14 This method sets the `System.Xml.XmlWriter.WriteState` to
15 `System.Xml.WriteState.Closed`.

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20

1 XmlWriter.Flush() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void Flush()  
4 [C#]  
5 public abstract void Flush()
```

6 Summary

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

8 Behaviors

9 As described above.

10

11 How and When to Override

12 This method must be overridden in order to provide the functionality described above,
13 as there is no default implementation.

14

15

1 XmlWriter.LookupPrefix(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract string LookupPrefix(string ns)  
5 [C#]  
6 public abstract string LookupPrefix(string ns)
```

7 Summary

8 Retrieves the prefix defined in the current namespace scope for the specified namespace
9 URI.

10 Parameters

Parameter	Description
<i>ns</i>	A <code>System.String</code> specifying the namespace URI.

11 Return Value

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

16 Behaviors

17 As described above.

18

19 How and When to Override

20 This method must be overridden in order to provide the functionality described above,
21 as there is no default implementation.

22

23 Exceptions

Exception	Condition
<code>System.ArgumentException</code>	<i>ns</i> is null or <code>System.String.Empty</code> .

24

1 2 `XmlWriter.WriteAttributes(System.Xml.XmlReader, System.Boolean)` Method 3

```
4 [ILAsm]  
5 .method public hidebysig virtual void WriteAttributes(class  
6 System.Xml.XmlReader reader, bool defattr)  
  
7 [C#]  
8 public virtual void WriteAttributes(XmlReader reader, bool defattr)
```

9 Summary

10 Writes the attributes found at the current position of a `System.Xml.XmlReader`.

11 Parameters

Parameter	Description
<i>reader</i>	A <code>System.Xml.XmlReader</code> from which to copy the attributes.
<i>defattr</i>	A <code>System.Boolean</code> where <code>true</code> specifies to copy the default attributes from <i>reader</i> ; otherwise, <code>false</code> .

12

13 Behaviors

14 If the reader is positioned on an `Element` or `XmlDeclaration` node, this method writes
15 all the contained attributes. If the reader is positioned on an `Attribute` node, this
16 method writes the current attribute, then the rest of the attributes until the closing tag
17 is reached. If the reader is positioned on any other node type, this method throws an
18 exception.

19

20 Default

21 This method positions the reader by calling its
22 `System.Xml.XmlReader.MoveToFirstAttribute` and
23 `System.Xml.XmlReader.MoveToNextAttribute` methods, and retrieves the value of the
24 attributes by calling the `System.Xml.XmlReader.ReadAttributeValue` method of the
25 reader.

26

27 How and When to Override

1 Override this method to remove any content that would invalidate the document.

2

3 **Usage**

4 Use this method to write all the attributes found at the current position.

5

6 **Exceptions**

Exception	Condition
System.ArgumentNullException	<i>reader</i> is null.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .
System.Xml.XmlException	<i>reader</i> is not positioned on a node of <code>System.Xml.XmlNodeTypeAttribute</code> , <code>Element</code> , or <code>XmlDeclaration</code> .

7

8

1 2 XmlWriter.WriteString(System.String, System.String, System.String, System.String) Method

```
5 [ILAsm]  
6 .method public hidebysig instance void WriteAttributeString(string prefix,  
7 string localName, string ns, string value)  
  
8 [C#]  
9 public void WriteAttributeString(string prefix, string localName, string  
10 ns, string value)
```

11 Summary

12 Writes a new attribute with the specified prefix, local name, namespace URI, and value.

13 Parameters

Parameter	Description
<i>prefix</i>	A System.String specifying the namespace prefix of the attribute.
<i>localName</i>	A System.String specifying the local name of the attribute.
<i>ns</i>	A System.String specifying the namespace URI of the attribute.
<i>value</i>	A System.String specifying the value of the attribute.

14 15 Description

16 This method calls the following methods in order to write a complete attribute:

```
17 System.Xml.XmlWriter.WriteStartAttribute(prefix, localName, ns)  
18  
19 System.Xml.XmlWriter.WriteString(value)  
20  
21  
22 System.Xml.XmlWriter.WriteEndAttribute()
```

23 Exceptions

Exception	Condition
System.ArgumentException	<i>localName</i> is null or System.String.Empty.

System.InvalidOperationException

The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Element.

1

2

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

XmlWriter.WriteAttributeString(System.String, System.String) Method

```
[ILAsm]  
.method public hidebysig instance void WriteAttributeString(string  
localName, string value)  
  
[C#]  
public void WriteAttributeString(string localName, string value)
```

Summary

Writes a new attribute with the specified local name and value.

Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the attribute.
<i>value</i>	A System.String specifying the value of the attribute.

Description

This method calls the following methods in order to write a complete attribute:

- System.Xml.XmlWriter.WriteStartAttribute(null, *localName*, null)
- System.Xml.XmlWriter.WriteString(*value*)
- System.Xml.XmlWriter.WriteEndAttribute()

Exceptions

Exception	Condition
System.ArgumentException	<i>localName</i> is null or System.String.Empty.
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Element.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

XmlWriter.WriteAttributeString(System.String, System.String, System.String) Method

```
[ILAsm]  
.method public hidebysig instance void WriteAttributeString(string  
localName, string ns, string value)  
  
[C#]  
public void WriteAttributeString(string localName, string ns, string  
value)
```

Summary

Writes a new attribute with the specified local name, namespace URI, and value.

Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the attribute.
<i>ns</i>	A System.String specifying the namespace URI of the attribute.
<i>value</i>	A System.String specifying the value of the attribute.

Description

This method calls the following methods in order to write a complete attribute:

- System.Xml.XmlWriter.WriteStartAttribute(null, *localName*, *ns*)
- System.Xml.XmlWriter.WriteString(*value*)
- System.Xml.XmlWriter.WriteEndAttribute()

Exceptions

Exception	Condition
System.ArgumentException	<i>localName</i> is null or System.String.Empty.
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Element.

1 XmlWriter.WriteBase64(System.Byte[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteBase64(class  
5 System.Byte[] buffer, int32 index, int32 count)  
  
6 [C#]  
7 public abstract void WriteBase64(byte[] buffer, int index, int count)
```

8 Summary

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

10 Parameters

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

11 12 Description

13 [Note: Base64 encoding represents byte sequences in a text form comprised of the 65
14 US-ASCII characters (A-Z, a-z, 0-9, +, /, =) where each character encodes 6 bits of the
15 binary data. For more information on Base64 encoding, see RFC 2045 (
16 <http://www.ietf.org/rfc/rfc2045>).]
17
18

19 Behaviors

20 As described above.

21

22 How and When to Override

23 This method must be overridden in order to provide the functionality described above,
24 as there is no default implementation.

25

1 Exceptions

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null.
System.ArgumentException	The buffer length minus <i>index</i> is less than <i>count</i> .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

2

3

1 XmlWriter.WriteBinHex(System.Byte[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteBinHex(class  
5 System.Byte[] buffer, int32 index, int32 count)  
  
6 [C#]  
7 public abstract void WriteBinHex(byte[] buffer, int index, int count)
```

8 Summary

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

10 Parameters

Parameter	Description
<i>buffer</i>	A <code>System.Byte</code> array containing the bytes to encode.
<i>index</i>	A <code>System.Int32</code> specifying the position within the array of the first byte to encode.
<i>count</i>	A <code>System.Int32</code> specifying the number of bytes to encode.

11 Description

12
13 [Note: For information on BinHex encoding, see RFC 1741
14 (<http://www.ietf.org/rfc/rfc1741>).]
15
16

17 Behaviors

18 As described above.
19

20 How and When to Override

21 This method must be overridden in order to provide the functionality described above,
22 as there is no default implementation.
23

24 Exceptions

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null.
System.ArgumentException	The buffer length minus <i>index</i> is less than <i>count</i> .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

1

2

1 XmlWriter.WriteCData(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteCData(string text)  
5 [C#]  
6 public abstract void WriteCData(string text)
```

7 Summary

8 Writes out a CDATA block containing the specified text.

9 Parameters

Parameter	Description
<i>text</i>	A <code>System.String</code> specifying the text to place inside the CDATA block.

10

11 Behaviors

12 This method writes `<![CDATA[text]>`.

13

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

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20 Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

21

1 XmlWriter.WriteCharEntity(System.Char)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteCharEntity(valuetype  
5 System.Char ch)  
  
6 [C#]  
7 public abstract void WriteCharEntity(char ch)
```

8 Summary

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

10 Parameters

Parameter	Description
<i>ch</i>	The <code>System.Char</code> for which to generate the entity.

11

12 Behaviors

13 This method forces the generation of a character entity for the specified Unicode
14 character value and writes the Unicode character in hexadecimal character entity
15 reference format.

16

17 How and When to Override

18 This method must be overridden in order to provide the functionality described above,
19 as there is no default implementation.

20

21 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.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1

2

1 XmlWriter.WriteChars(System.Char[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteChars(char[] buffer,  
5 int32 index, int32 count)  
  
6 [C#]  
7 public abstract void WriteChars(char[] buffer, int index, int count)
```

8 Summary

9 Writes text a buffer at a time.

10 Parameters

Parameter	Description
<i>buffer</i>	A <code>System.Char</code> array containing the text to write.
<i>index</i>	A <code>System.Int32</code> specifying the position within the array of the start of the text to write.
<i>count</i>	A <code>System.Int32</code> specifying the number of characters to write.

11

12 Behaviors

13 As described above.

14

15 How and When to Override

16 When overriding this method, throw an exception if surrogate pair characters would be
17 split across multiple buffer writes. This exception must be caught in order to continue
18 writing the next surrogate pair characters. The XML specification defines the valid
19 ranges for surrogate pairs.

20

21 This method must be overridden in order to provide the functionality described above,
22 as there is no default implementation.

23

24 Exceptions

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null.
System.ArgumentException	The buffer length minus <i>index</i> is less than <i>count</i> .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

1

2

1 XmlWriter.WriteComment(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteComment(string text)  
5 [C#]  
6 public abstract void WriteComment(string text)
```

7 Summary

8 Writes a comment containing the specified text.

9 Parameters

Parameter	Description
<i>text</i>	A <code>System.String</code> containing the text to place inside the comment.

10

11 Behaviors

12 This method writes `<!--text-->`.

13

14 If *text* is null or `System.String.Empty`, this method writes a comment with no content,
15 `<!-->`.

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20 Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

21

1 XmlWriter.WriteDocType(System.String, 2 System.String, System.String, System.String) 3 Method

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract void WriteDocType(string name,  
6 string pubid, string sysid, string subset)  
  
7 [C#]  
8 public abstract void WriteDocType(string name, string pubid, string sysid,  
9 string subset)
```

10 Summary

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

12 Parameters

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

13

14 Behaviors

15 This method writes the document type declaration with the specified name and optional
16 attributes. The optional attributes, *pubid*, *sysid*, and *subset*, are not checked for invalid
17 characters.

18

19 How and When to Override

20 This method must be overridden in order to provide the functionality described above,
21 as there is no default implementation.

22

1 Exceptions

Exception	Condition
System.ArgumentException	<i>name</i> is null or <code>System.String.Empty</code> . -or- The value for <i>name</i> would result in invalid XML.
System.InvalidOperationException	This method was called outside the prolog (after the root element).

2

3

1
2 **XmlWriter.WriteString(System.String**
3 **, System.String, System.String) Method**

```
4 [ILAsm]  
5 .method public hidebysig instance void WriteElementString(string  
6 localName, string ns, string value)  
  
7 [C#]  
8 public void WriteElementString(string localName, string ns, string value)
```

9 **Summary**

10 Writes an element with the specified local name, namespace URI, and value.

11 **Parameters**

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the element.
<i>ns</i>	A System.String specifying the namespace URI to associate with the element.
<i>value</i>	A System.String specifying the value of the element.

12
13 **Description**

14 This method calls the following methods to write a complete element:

15
16 System.Xml.XmlWriter.WriteStartElement(*localName*, *ns*)

17
18 System.Xml.XmlWriter.WriteString(*value*) - this method is not called if *value* is either
19 null or System.String.Empty

20
21 System.Xml.XmlWriter.WriteEndElement()

22
23 If any of the input parameters are null or System.String.Empty, the element is written
24 with that parameter missing.

25 **Exceptions**

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1

2

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

XmlWriter.WriteString(System.String, System.String) Method

```
[ILAsm]  
.method public hidebysig instance void WriteElementString(string  
localName, string value)  
  
[C#]  
public void WriteElementString(string localName, string value)
```

Summary

Writes an element with the specified local name and value.

Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the element.
<i>value</i>	A System.String specifying the value of the element.

Description

This method is equivalent to System.Xml.XmlWriter.WriteString(*localName*, null, *value*).

Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1 XmlWriter.WriteEndElement() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteEndElement()  
4 [C#]  
5 public abstract void WriteEndElement()
```

6 Summary

7 Closes the attribute started with the System.Xml.XmlWriter.WriteStartElement
8 method.

9 Behaviors

10 As described above.

11

12 How and When to Override

13 Override the System.Xml.XmlWriter.WriteStartElement,
14 System.Xml.XmlWriter.WriteAttributeString, and
15 System.Xml.XmlWriter.WriteEndElement methods so these methods also close any
16 attributes that are open when they are called.

17

18 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Attribute.

19

20

1 XmlWriter.WriteEndElement() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteEndElement()  
4 [C#]  
5 public abstract void WriteEndElement()
```

6 Summary

7 Closes open elements and attributes and sets the `System.Xml.XmlWriter.WriteState`
8 back to the `System.Xml.WriteState.Start` state.

9 Behaviors

10 This method closes all elements and attributes created by the
11 `System.Xml.XmlWriter.WriteStartElement` and
12 `System.Xml.XmlWriter.WriteStartAttribute` methods, respectively, that are open
13 when the `System.Xml.XmlWriter.WriteEndElement` method is called.

14

15 How and When to Override

16 This method must be overridden in order to provide the functionality as described
17 herein, as there is no default implementation.

18

19 Usage

20 After calling this method, the current instance can be used to write a new XML
21 document.

22

23 Exceptions

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

24

25

1 XmlWriter.WriteEndElement() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteEndElement()  
4 [C#]  
5 public abstract void WriteEndElement()
```

6 Summary

7 Closes an open element and pops the corresponding namespace scope.

8 Behaviors

9 This method closes one element and pops the corresponding namespace scope. If the
10 open element does not contain content, it is closed as an empty element using " />";
11 otherwise an end element is written.

12

13 How and When to Override

14 This method must be overridden in order to provide the functionality described above,
15 as there is no default implementation.

16

17 Exceptions

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

18

19

1 XmlWriter.WriteEntityRef(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteEntityRef(string name)  
5 [C#]  
6 public abstract void WriteEntityRef(string name)
```

7 Summary

8 Writes an entity reference with the specified name.

9 Parameters

Parameter	Description
<i>name</i>	A <code>System.String</code> specifying the name of the entity reference.

10

11 Behaviors

12 This method writes `%name;`.

13

14 How and When to Override

15 This method must be overridden in order to provide the functionality described above,
16 as there is no default implementation.

17

18 Exceptions

Exception	Condition
System.ArgumentException	<i>name</i> is either null or <code>System.String.Empty</code> .

19

20

1 XmlWriter.WriteEndElement() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteEndElement()  
4 [C#]  
5 public abstract void WriteEndElement()
```

6 Summary

7 Closes an open element and pops the corresponding namespace scope.

8 Behaviors

9 This method closes one element and pops the corresponding namespace scope. This
10 method writes an end element regardless of whether there is any content in the
11 element.

12

13 How and When to Override

14 This method must be overridden in order to provide the functionality described above,
15 as there is no default implementation.

16

17 Exceptions

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

18

19

1 XmlWriter.WriteName(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteName(string name)  
5 [C#]  
6 public abstract void WriteName(string name)
```

7 Summary

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

10 Parameters

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

11

12 Behaviors

13 This method writes the specified name, ensuring it is a valid name according to the W3C
14 XML 1.0 recommendation.

15

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20 Exceptions

Exception	Condition
System.ArgumentException	<i>name</i> is null or <code>System.String.Empty</code> ; or <i>name</i> is not a valid XML Name.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

1

2

1 XmlWriter.WriteNmToken(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteNmToken(string name)  
5 [C#]  
6 public abstract void WriteNmToken(string name)
```

7 Summary

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

11 Parameters

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

12

13 Behaviors

14 As described above.

15

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20 Exceptions

Exception	Condition
System.ArgumentException	<i>name</i> is null or <code>System.String.Empty</code> ; or <i>name</i> is not a valid XML Nmtoken.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

1

2

1 2 XmlWriter.WriteNode(System.Xml.XmlReader 3 , System.Boolean) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual void WriteNode(class System.Xml.XmlReader  
6 reader, bool defattr)  
7 [C#]  
8 public virtual void WriteNode(XmlReader reader, bool defattr)
```

9 Summary

10 Writes the node found at the current position of the specified `System.Xml.XmlReader`,
11 and all sub-nodes.

12 Parameters

Parameter	Description
<i>reader</i>	A <code>System.Xml.XmlReader</code> from which to copy the attributes.
<i>defattr</i>	A <code>System.Boolean</code> where <code>true</code> specifies to copy the default attributes from <i>reader</i> ; otherwise, <code>false</code> .

13

14 Behaviors

15 After the nodes are read, *reader* is moved to the next node at the same depth as the
16 current node.

17

18 Default

19 If *reader* is in the `System.Xml.ReadState.InitialSystem.Xml.XmlReader.ReadState`,
20 this method moves *reader* to the end of the stream. If *reader* is in the
21 `System.Xml.ReadState.EndOfFileSystem.Xml.XmlReader.ReadState`, this method is
22 non-operational.

23

24 How and When to Override

25 Override this method to customize the behavior of this method in types derived from the
26 `System.Xml.XmlWriter` class.

1

2 Usage

3 Use this method to write the node found at the current position and all sub-nodes.

4

5 Exceptions

Exception	Condition
System.ArgumentNullException	<i>reader</i> is null.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

6

7 Example

8 The following example uses a `System.Xml.XmlTextReader` and a
9 `System.Xml.XmlTextWriter` to copy an XML file, specified in the command line, to the
10 console.

11

12

```

13 using System;
14 using System.Xml;
15
16 public class Copier {
17
18     public static void Main(string[] args) {
19
20         XmlTextReader xtReader = new XmlTextReader(args[0]);
21         XmlTextWriter xtWriter =
22             new XmlTextWriter(Console.Out);
23         xtWriter.WriteNode(xtReader, false);
24         xtWriter.Close();
25         xtReader.Close();
26     }
27 }
28

```

1 2 XmlWriter.WriteProcessingInstruction(System.String, System.String) Method 3

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract void  
6 WriteProcessingInstruction(string name, string text)  
7  
8 [C#]  
9 public abstract void WriteProcessingInstruction(string name, string text)
```

9 Summary

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

11 Parameters

Parameter	Description
<i>name</i>	A System.String specifying the name of the processing instruction.
<i>text</i>	A System.String specifying the text to include in the processing instruction.

12

13 Behaviors

14 This method writes `<?name?text?>`.

15

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

18 How and When to Override

19 This method must be overridden in order to provide the functionality described above,
20 as there is no default implementation.

21

22 Exceptions

Exception	Condition
System.ArgumentException	The text would result in a non-well formed XML document.

	<p>- or -</p> <p><i>name</i> is null or <code>System.String.Empty</code>.</p> <p>- or -</p> <p>This method is being used to create an XML declaration after <code>System.Xml.XmlWriter.WriteStartDocument</code> has already been called.</p>
System.InvalidOperationException	<p>The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code>.</p>

1

2

1
2 **XmlWriter.WriteQualifiedName(System.String**
3 **, System.String) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract void WriteQualifiedName(string  
6 localName, string ns)  
  
7 [C#]  
8 public abstract void WriteQualifiedName(string localName, string ns)
```

9 **Summary**

10 Writes the qualified name.

11 **Parameters**

Parameter	Description
<i>localName</i>	A System.String specifying the local name to write.
<i>ns</i>	A System.String specifying the namespace URI to associate with <i>localname</i> .

12

13 **Behaviors**

14 This method writes the qualified name. If *ns* maps to the current default namespace, no
15 prefix is generated. When writing attribute values, this method generates a prefix if *ns* is
16 not found. When writing element content, it throws an exception if *ns* is not found.

17

18 **How and When to Override**

19 This method must be overridden in order to provide the functionality described above,
20 as there is no default implementation.

21

22 **Exceptions**

Exception	Condition
System.ArgumentException	<i>localName</i> is null, is System.String.Empty, or is

	not a valid XML name.
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1

2

1 XmlWriter.WriteRaw(System.Char[], 2 System.Int32, System.Int32) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteRaw(char[] buffer,  
5 int32 index, int32 count)  
  
6 [C#]  
7 public abstract void WriteRaw(char[] buffer, int index, int count)
```

8 Summary

9 Writes raw text from a character array.

10 Parameters

Parameter	Description
<i>buffer</i>	A <code>System.Char</code> array containing the text to write.
<i>index</i>	A <code>System.Int32</code> specifying the position within the array of the start of the text to write.
<i>count</i>	A <code>System.Int32</code> specifying the number of characters to write.

11

12 Behaviors

13 This method writes raw text from a character array. This method does not encode any
14 characters.

15

16 How and When to Override

17 This method must be overridden in order to provide the functionality described above,
18 as there is no default implementation.

19

20 Exceptions

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

System.ArgumentNullException	<i>buffer</i> is null.
System.ArgumentException	The buffer length minus <i>index</i> is less than <i>count</i> .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero.
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1

2

1 XmlWriter.WriteRaw(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteRaw(string data)  
4 [C#]  
5 public abstract void WriteRaw(string data)
```

6 Summary

7 Writes raw text from a string.

8 Parameters

Parameter	Description
<i>data</i>	A System.String specifying the text to write.

9

10 Behaviors

11 This method writes raw text from a string. This method does not encode any characters.
12 If *data* is null, System.String.Empty is written.

13

14 How and When to Override

15 This method must be overridden in order to provide the functionality described above,
16 as there is no default implementation.

17

18 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

19

20

1
2 **XmlWriter.WriteStartAttribute(System.String,**
3 **System.String, System.String) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract void WriteStartAttribute(string  
6 prefix, string localName, string ns)  
  
7 [C#]  
8 public abstract void WriteStartAttribute(string prefix, string localName,  
9 string ns)
```

10 **Summary**

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

13 **Parameters**

Parameter	Description
<i>prefix</i>	A System.String specifying the namespace prefix of the attribute.
<i>localName</i>	A System.String specifying the local name of the attribute.
<i>ns</i>	A System.String specifying the namespace URI associated with the attribute.

14
15 **Behaviors**
16 If any of the input parameters are null or System.String.Empty, the start attribute is
17 written with that parameter missing.

18 **How and When to Override**

19 When overriding this method, close any open attributes before writing the new attribute.
20
21 This method must be overridden in order to provide the functionality described above,
22 as there is no default implementation.

23
24 **Exceptions**

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

System.InvalidOperationException

The System.Xml.XmlWriter.WriteState is not one of the following: System.Xml.WriteState.Attribute or System.Xml.WriteState.Element.

1

2

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

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

```
[ILAsm]  
.method public hidebysig instance void WriteStartAttribute(string  
localName, string ns)  
  
[C#]  
public void WriteStartAttribute(string localName, string ns)
```

Summary

Writes the start of an attribute.

Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the attribute.
<i>ns</i>	A System.String specifying the namespace URI of the attribute.

Description

This method calls System.Xml.XmlWriter.WriteStartAttribute(null, localName, ns).

Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1
2 **XmlWriter.WriteStartDocument(System.Boolean) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig virtual abstract void WriteStartDocument(bool  
6 standalone)  
7 [C#]  
8 public abstract void WriteStartDocument(bool standalone)
```

9 **Summary**

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

11 **Parameters**

Parameter	Description
<i>standalone</i>	A System.Boolean where true indicates to write "standalone= yes"; false indicates to write "standalone=no".

12
13 **Description**

14 [Note: Character encoding is set when the class is instantiated.
15
16]

17 **Behaviors**

18 As described above.

19

20 **How and When to Override**

21 This method must be overridden in order to provide the functionality described above,
22 as there is no default implementation.

23

24 **Exceptions**

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

System.InvalidOperationException

The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Start.

1

2

1 XmlWriter.WriteStartDocument() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract void WriteStartDocument()  
4 [C#]  
5 public abstract void WriteStartDocument()
```

6 Summary

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

8 Description

9 *[Note:* Character encoding is set when the class is instantiated.
10
11]

12 Behaviors

13 As described above.

14

15 How and When to Override

16 This method must be overridden in order to provide the functionality described above,
17 as there is no default implementation.

18

19 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is not System.Xml.WriteState.Start.

20

21

1 XmlWriter.WriteStartElement(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig instance void WriteStartElement(string localName)  
5 [C#]  
6 public void WriteStartElement(string localName)
```

7 Summary

8 Writes a start element with the specified name.

9 Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the element.

10 11 Description

12 This method calls System.Xml.XmlWriter.WriteStartElement(null, *localName*, null).

13 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

14
15

1 XmlWriter.WriteStartElement(System.String, 2 System.String) Method

```
3 [ILAsm]  
4 .method public hidebysig instance void WriteStartElement(string localName,  
5 string ns)  
  
6 [C#]  
7 public void WriteStartElement(string localName, string ns)
```

8 Summary

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

11 Parameters

Parameter	Description
<i>localName</i>	A System.String specifying the local name of the element.
<i>ns</i>	A System.String specifying the namespace URI to associate with the element.

12 13 Description

14 This method calls System.Xml.XmlWriter.WriteStartElement(null, localName, ns).

15 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

16
17

1 XmlWriter.WriteStartElement(System.String, 2 System.String, System.String) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteStartElement(string  
5 prefix, string localName, string ns)  
  
6 [C#]  
7 public abstract void WriteStartElement(string prefix, string localName,  
8 string ns)
```

9 Summary

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

12 Parameters

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

14 Behaviors

15 This method writes a start element and name, and associates it with a namespace and
16 prefix. If the namespace is already in scope and has an associated prefix, that prefix is
17 automatically written also.

18
19 If any of the input parameters are null or System.String.Empty, the start element is
20 written with that parameter missing.

21 How and When to Override

22 When overriding this method, also override the
23 System.Xml.XmlWriter.WriteEndElement, System.Xml.XmlWriter.WriteString, and
24 System.Xml.XmlWriter.WriteRaw methods so they close any open start element.

26 Usage

1 Use this method to write a specified start element and name, and associate it with a
2 given namespace and prefix. Write any attributes using the
3 `System.Xml.XmlWriter.WriteStartAttribute`, `System.Xml.XmlWriter.WriteString`,
4 and `System.Xml.XmlWriter.WriteEndAttribute` methods, then close the element
5 using the `System.Xml.XmlWriter.WriteEndElement` method.

6

7 Exceptions

Exception	Condition
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

8

9 Example

10 See `System.Xml.XmlTextWriter.WriteStartElement` for an example demonstrating
11 this method.

12

1 XmlWriter.WriteString(System.String)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteString(string text)  
5 [C#]  
6 public abstract void WriteString(string text)
```

7 Summary

8 Writes the specified text.

9 Parameters

Parameter	Description
<i>text</i>	A <code>System.String</code> specifying the text to write.

10

11 Behaviors

12 This method performs the following conversions before writing the text:

- 13 • The characters '&', '<', and '>' are replaced with "&", "<", and ">",
14 respectively.
- 15 • Character values in the range 0x-0x1F (excluding the white space characters 0x9,
16 0x10, and 0x13) are replaced with numeric character entities ("�" through
17 "�x1F").
- 18 • If called in the context of an attribute value, double and single quotes are replaced
19 with """ and "'" respectively.

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

22 How and When to Override

23 This method must be overridden in order to provide the functionality described above,
24 as there is no default implementation.

25

26 Exceptions

Exception	Condition
System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed.

1

2

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

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

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

Summary

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

Parameters

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

Description

[Note: Applications encode DOM strings using UTF-16. 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 might therefore in some cases correspond to two 16-bit units in a DOM 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>.

]

Behaviors

This method generates and writes the surrogate character entity for a surrogate character pair. The surrogate character entity is written in hexadecimal format. The range for surrogate characters is #x10000 to #x10FFFF. The following formula is used

1 to generate the surrogate character entity: $(highChar - 0xD800) * 0x400 + (lowChar$
2 $- 0xDC00) + 0x10000$.

3

4 **How and When to Override**

5 This method must be overridden in order to provide the functionality described above,
6 as there is no default implementation.

7

8 **Exceptions**

Exception	Condition
System.ArgumentException	An invalid surrogate character pair was passed.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

9

10

1 XmlWriter.WriteWhitespace(System.String) 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual abstract void WriteWhitespace(string ws)  
5 [C#]  
6 public abstract void WriteWhitespace(string ws)
```

7 Summary

8 Writes the given white space.

9 Parameters

Parameter	Description
<i>ws</i>	A <code>System.String</code> containing the white space characters.

10

11 Behaviors

12 As described above.

13

14 How and When to Override

15 This method must be overridden in order to provide the functionality described above,
16 as there is no default implementation.

17

18 Exceptions

Exception	Condition
System.ArgumentException	<i>ws</i> is null or <code>System.String.Empty</code> or contains non-white space characters.
System.InvalidOperationException	The <code>System.Xml.XmlWriter.WriteState</code> is <code>System.Xml.WriteState.Closed</code> .

19

20

1 XmlWriter.WriteState Property

```
2 [ILAsm]  
3 .property valuetype System.Xml.WriteState WriteState { public hidebysig  
4 virtual abstract specialname valuetype System.Xml.WriteState  
5 get_WriteState() }  
6 [C#]  
7 public abstract WriteState WriteState { get; }
```

8 Summary

9 Gets the write state of the writer.

10 Property Value

11 One of the members of the `System.Xml.WriteState` enumeration.

12 Behaviors

13 As described above.

14

15 This property is read-only.

16 How and When to Override

17 This property must be overridden in order to provide the functionality as described
18 herein, as there is no default implementation.

19

20 Usage

21 Use this property to query the current state, for example, whether the writer is newly
22 initialized, writing specific XML structures, or closed.

23

24

1 XmlWriter.XmlLang Property

```
2 [ILAsm]  
3 .property string XmlLang { public hidebysig virtual abstract specialname  
4 string get_XmlLang() }  
5 [C#]  
6 public abstract string XmlLang { get; }
```

7 Summary

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

10 Property Value

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

13 Behaviors

14 As described above.

15
16 This property is read-only.

17 How and When to Override

18 This property must be overridden in order to provide the functionality described above,
19 as there is no default implementation.

20

21

1 XmlWriter.XmlSpace Property

```
2 [ILAsm]  
3 .property valuetype System.Xml.XmlSpace XmlSpace { public hidebysig  
4 virtual abstract specialname valuetype System.Xml.XmlSpace get_XmlSpace()  
5 }  
6 [C#]  
7 public abstract XmlSpace XmlSpace { get; }
```

8 Summary

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

11 Property Value

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

14 Behaviors

15 As described above.

16
17 This property is read-only.

18 How and When to Override

19 This property must be overridden in order to provide the functionality described above,
20 as there is no default implementation.

21

22