

1 System.Xml.XmlWriter Class

2
3

```
4 [ILASM]  
5 .class public abstract XmlWriter extends System.Object  
6 [C#]  
7 public abstract class XmlWriter
```

8 Assembly Info:

- 9 • Name: System.Xml
- 10 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 11 • Version: 1.0.x.x
- 12 • Attributes:
 - 13 ○ CLSCompliantAttribute(true)

14 Summary

15

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

18 Inherits From: System.Object

19

20 **Library:** XML

21

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

24

25 Description

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

28

29 [Note: This class is **abstract** and is implemented in the
30 **System.Xml.XmlTextWriter** class.]

31

1 XmlWriter() Constructor

```
2 [ILASM]  
3 family 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
12 the underlying 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
18 described above, as there is no default implementation.

19

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
8 underlying stream.

9 Behaviors

10 As described above.

11 How and When to Override

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

14

1 XmlWriter.LookupPrefix(System.String)

2 Method

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

8 Summary

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

11 Parameters

12
13

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

14
15
16

15 Return Value

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

21 Behaviors

22 As described above.

23 How and When to Override

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

26 Exceptions

27
28

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

29
30
31

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

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

9 Summary

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

12 Parameters

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

17 Behaviors

18 If the reader is positioned on an **Element** or **XmlDeclaration** node,
19 this method writes all the contained attributes. If the reader is
20 positioned on an **Attribute** node, this method writes the current
21 attribute, then the rest of the attributes until the closing tag is
22 reached. If the reader is positioned on any other node type, this
23 method throws an exception.

24 Default

25 This method positions the reader by calling its
26 **System.Xml.XmlReader.MoveToFirstAttribute** and
27 **System.Xml.XmlReader.MoveToNextAttribute** methods, and
28 retrieves the value of the attributes by calling the
29 **System.Xml.XmlReader.ReadAttributeValue** method of the reader.

30 How and When to Override

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

1 **Usage**

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

4 **Exceptions**

5
6

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

7
8
9

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

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

11 Summary

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

14 Parameters

15
16

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.

17
18

18 Description

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

21
22
23
24
25
26

System.Xml.XmlWriter.WriteStartAttribute(*null*, *localName*, *ns*)

System.Xml.XmlWriter.WriteString(*value*)

System.Xml.XmlWriter.WriteEndAttribute()

27 Exceptions

28
29

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

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

```
3 [ILASM]  
4 .method public hidebysig instance void  
5 WriteAttributeString(string localName, string value)  
  
6 [C#]  
7 public void WriteAttributeString(string localName, string  
8 value)
```

9 Summary

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

11 Parameters

12
13

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.

14
15

15 Description

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

18
19
20
21
22
23
24

System.Xml.XmlWriter.XmlWriter.WriteStartAttribute(**null**,
localName, **null**)

System.Xml.XmlWriter.WriteString(*value*)

System.Xml.XmlWriter.WriteEndAttribute()

25 Exceptions

26
27

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 .

28
29
30

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

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

11 Summary

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

14 Parameters

15
16

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.

17
18

18 Description

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

21
22
23
24
25
26

System.Xml.XmlWriter.WriteStartAttribute(*prefix*, *localName*, *ns*)

System.Xml.XmlWriter.WriteString(*value*)

System.Xml.XmlWriter.WriteEndAttribute()

27 Exceptions

28
29

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

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

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

10 Summary

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

13 Parameters

14
15

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.

16
17

17 Description

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

22 Behaviors

23 As described above.

24 How and When to Override

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

27 Exceptions

28
29

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
3

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

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

10 Summary

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

13 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.

16 Description

18 [Note: For information on BinHex encoding, see RFC 1741
19 (<http://www.ietf.org/rfc/rfc1741>).]

20 Behaviors

21 As described above.

22 How and When to Override

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

25 Exceptions

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentException	The buffer length minus <i>index</i> is less

- 1
- 2
- 3

	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 XmlWriter.WriteCData(System.String)

2 Method

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

8 Summary

9 Writes out a CDATA block containing the specified text.

10 Parameters

11
12

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

13
14

15 Behaviors

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

17

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

20 How and When to Override

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

23 Exceptions

24
25

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

26
27
28

1 XmlWriter.WriteCharEntity(System.Char) 2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual abstract void  
5 WriteCharEntity(valuetype 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
10 character value.

11 Parameters

12
13

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

14
15

16 Behaviors

17 This method forces the generation of a character entity for the
18 specified Unicode character value and writes the Unicode character in
19 hexadecimal character entity reference format.

20 How and When to Override

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

23 Exceptions

24
25

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 .

26
27
28

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

```
3 [ILASM]  
4 .method public hidebysig virtual abstract void  
5 WriteChars(class System.Char[] buffer, int32 index, int32  
6 count)  
  
7 [C#]  
8 public abstract void WriteChars(char[] buffer, int index,  
9 int count)
```

10 Summary

11 Writes text a buffer at a time.

12 Parameters

13
14

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

15
16

17 Behaviors

18 As described above.

19 How and When to Override

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

25 Exceptions

26
27

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentException	The buffer length minus <i>index</i> is less than <i>count</i> .

1
2
3

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 XmlWriter.WriteComment(System.String)

2 Method

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

8 Summary

9 Writes a comment containing the specified text.

10 Parameters

11
12

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

13
14

15 Behaviors

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

17

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

20 How and When to Override

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

23 Exceptions

24
25

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

26
27
28

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

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

11 Summary

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

14 Parameters

15
16

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.

17
18

19 Behaviors

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

23 How and When to Override

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

1 **Exceptions**
2
3

Exception	Condition
System.ArgumentException	<i>name</i> is null or System.String.Empty . -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).

4
5
6

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

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

9 Summary

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

11 Parameters

12
13

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.

14
15

Description

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

19 Exceptions

20
21

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

22
23
24

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

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

11 Summary

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

14 Parameters

15
16

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.

17
18

Description

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

20

21 **System.Xml.XmlWriter.WriteStartElement**(*localName*, *ns*)

22

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

25

26 **System.Xml.XmlWriter.WriteEndElement**()

27

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

30 Exceptions

31

32

1
2
3

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  
4 WriteEndElement()  
5  
6 [C#]  
7 public abstract void WriteEndElement()
```

7 Summary

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

10 Behaviors

11 As described above.

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
16 methods also close any attributes that are open when they are called.

17 Exceptions

18
19

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

20
21
22

1 XmlWriter.WriteEndElement() Method

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

7 Summary

8 Closes open elements and attributes and sets the
9 **System.Xml.XmlWriter.WriteState** back to the
10 **System.Xml.WriteState.Start** state.

11 Behaviors

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

17 How and When to Override

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

20 Usage

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

23 Exceptions

24
25

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

26
27
28

1 XmlWriter.WriteEndElement() Method

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

7 Summary

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

9 Behaviors

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

14 How and When to Override

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

17 Exceptions

18
19

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

20
21
22

1 XmlWriter.WriteEntityRef(System.String)

2 Method

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

8 Summary

9 Writes an entity reference with the specified name.

10 Parameters

11
12

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

13
14

15 Behaviors

16 This method writes % *name*;

17 How and When to Override

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

20 Exceptions

21
22

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

23
24
25

1 XmlWriter.WriteEndElement() Method

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

7 Summary

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

9 Behaviors

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

13 How and When to Override

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

16 Exceptions

17
18

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

19
20
21

1 XmlWriter.WriteName(System.String)

2 Method

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

8 Summary

9 Writes the specified name, ensuring it is a valid name according to the
10 W3C XML 1.0 recommendation (<http://www.w3.org/TR/1998/REC-xml-19980210#NT-Name>).
11

12 Parameters

13
14

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

15
16

17 Behaviors

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

20 How and When to Override

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

23 Exceptions

24
25

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

26
27
28

1 XmlWriter.WriteNmToken(System.String)

2 Method

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

8 Summary

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

12 Parameters

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

15
16

17 Behaviors

18 As described above.

19 How and When to Override

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

22 Exceptions

23
24

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

25
26
27

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

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

9 Summary

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

12 Parameters

13
14

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

15
16

17 Behaviors

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

20 Default

21 If *reader* is in the **System.Xml.ReadState.Initial**
22 **System.Xml.XmlReader.ReadState**, this method moves *reader* to
23 the end of the stream. If *reader* is in the
24 **System.Xml.ReadState.EndOfFile**
25 **System.Xml.XmlReader.ReadState**, this method is non-operational.

26 How and When to Override

27 Override this method to customize the behavior of this method in
28 types derived from the **System.Xml.XmlWriter** class.

29 Usage

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

3 Exceptions

4
5

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

6
7
8

Example

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

12
13

[C#]

14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

```
using System;  
using System.Xml;  
  
public class Copier {  
    public static void Main(string[] args) {  
        XmlTextReader xtReader = new XmlTextReader(args[0]);  
        XmlTextWriter xtWriter =  
            new XmlTextWriter(Console.Out);  
        xtWriter.WriteNode(xtReader, false);  
        xtWriter.Close();  
        xtReader.Close();  
    }  
}
```

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

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

9 Summary

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

11 Parameters

12
13

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.

14
15

16 Behaviors

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

18

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

21 How and When to Override

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

24 Exceptions

25
26

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

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

1
2
3

1 XmlWriter.WriteQualifiedName(System.String, System.String) Method

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

9 Summary

10 Writes the qualified name.

11 Parameters

12
13

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> .

14
15

16 Behaviors

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

21 How and When to Override

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

24 Exceptions

25
26

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

1
2
3

1 XmlWriter.WriteRaw(System.String)

2 Method

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

8 Summary

9 Writes raw text from a string.

10 Parameters

11
12

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

13
14

15 Behaviors

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

19 How and When to Override

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

22 Exceptions

23
24

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

25
26
27

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

```
3 [ILASM]  
4 .method public hidebysig virtual abstract void  
5 WriteRaw(class System.Char[] buffer, int32 index, int32  
6 count)  
  
7 [C#]  
8 public abstract void WriteRaw(char[] buffer, int index, int  
9 count)
```

10 Summary

11 Writes raw text from a character array.

12 Parameters

13
14

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

15
16

17 Behaviors

18 This method writes raw text from a character array. This method does
19 not encode any characters.

20 How and When to Override

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

23 Exceptions

24
25

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.

1
2
3

System.InvalidOperationException	The System.Xml.XmlWriter.WriteState is System.Xml.WriteState.Closed .
---	---

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

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

11 Summary

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

14 Parameters

15
16

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.

17
18

19 Behaviors

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

22 How and When to Override

23 When overriding this method, close any open attributes before writing
24 the new attribute.

25 Exceptions

26
27

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

1
2
3

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

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

9 Summary

10 Writes the start of an attribute.

11 Parameters

12
13

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.

14
15

15 Description

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

18 Exceptions

19
20

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

21
22
23

1 XmlWriter.WriteStartDocument(System.Boolean) Method

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

8 Summary

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

11 Parameters

12
13

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

14
15

15 Description

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

17 Behaviors

18 As described above.

19 How and When to Override

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

22 Exceptions

23
24

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

25
26
27

1 XmlWriter.WriteStartDocument() Method

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

7 Summary

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

10 Description

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

12 Behaviors

13 As described above.

14 How and When to Override

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

17 Exceptions

18
19

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

20
21
22

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

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

10 Summary

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

13 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.

16
17

18 Behaviors

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

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

25 How and When to Override

26 When overriding this method, also override the
27 **System.Xml.XmlWriter.WriteEndElement**,
28 **System.Xml.XmlWriter.WriteString**, and
29 **System.Xml.XmlWriter.WriteRaw** methods so they close any open
30 start element.

1 **Usage**

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

9 **Exceptions**

10
11

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

12
13
14

Example

15 See **System.Xml.XmlTextWriter.WriteStartElement** for an
16 example demonstrating this method.

17

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

```
3 [ILASM]  
4 .method public hidebysig instance void  
5 WriteStartElement(string localName, 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
10 the given namespace.

11 Parameters

12
13

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.

14
15

15 Description

16 This method calls **System.Xml.XmlWriter.WriteStartElement(null,**
17 *localName, ns*).

18 Exceptions

19
20

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

21
22
23

1 XmlWriter.WriteStartElement(System.String) Method

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

8 Summary

9 Writes a start element with the specified name.

10 Parameters

11
12

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

13
14

14 Description

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

17 Exceptions

18
19

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

20
21
22

1 XmlWriter.WriteString(System.String)

2 Method

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

8 Summary

9 Writes the specified text.

10 Parameters

11
12

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

13
14

15 Behaviors

16 This method performs the following conversions before writing the
17 text:

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

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

27 How and When to Override

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

1 **Exceptions**

2

3

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

4

5

6

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

```
3 [ILASM]
4 .method public hidebysig virtual abstract void
5 WriteSurrogateCharEntity(valuetype System.Char lowChar,
6 valuetype System.Char highChar)
7
8 [C#]
9 public abstract void WriteSurrogateCharEntity(char lowChar,
char highChar)
```

10 Summary

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

13 Parameters

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

17 Description

18 [Note: Applications encode DOM strings using UTF-16. For both HTML
19 and XML, the document character set (and therefore the notation of
20 numeric character references) is based on UCS [ISO-10646]. A single
21 numeric character reference in a source document may therefore in
22 some cases correspond to two 16-bit units in a DOM string (a high
23 surrogate and a low surrogate). These 16-bit units are referred to as a
24 surrogate pair.

25
26 For more information regarding surrogates or characters, refer to
27 section 3.7 of the Unicode 3.0/Unicode 2.0 standard located at
28 <http://www.unicode.org>, or section 2.2 of the W3C XML 1.0
29 Recommendation located at <http://www.w3.org/TR/REC-xml#charsets>.]

31 Behaviors

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

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

3 How and When to Override

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

6 Exceptions

7

8

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

9

10

11

1 XmlWriter.WriteWhitespace(System.String) Method

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

8 Summary

9 Writes the given white space.

10 Parameters

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

15 Behaviors

16 As described above.

17 How and When to Override

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

20 Exceptions

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

23
24
25

1 XmlWriter.WriteState Property

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

8 Summary

9 Gets the write state of the writer.

10 Property Value

11

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

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
19 as described herein, as there is no default implementation.

20 Usage

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

23

1 XmlWriter.XmlLang Property

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

7 Summary

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

11 Property Value

12

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

15 Behaviors

16 As described above.

17

18 This property is read-only.

19 How and When to Override

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

22

1 XmlWriter.XmlSpace Property

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

8 Summary

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

11 Property Value

12

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

16 Behaviors

17 As described above.

18

19 This property is read-only.

20 How and When to Override

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

23