

1 System.IO.StreamWriter Class

```
2 [ILAsm]  
3 .class public serializable StreamWriter extends System.IO.TextWriter  
4 [C#]  
5 public class StreamWriter: TextWriter
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 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 Implements:

- 13 • **System.IDisposable**

14 Summary

15 Implements a `System.IO.Stream` wrapper that writes characters to a stream in a
16 particular encoding.

17 Inherits From: System.IO.TextWriter

18
19 **Library:** BCL

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

24 Description

25 The `System.IO.StreamWriter` class is designed for character output in a particular
26 `System.Text.Encoding`, whereas subclasses of `System.IO.Stream` are designed for
27 byte input and output.

28
29 `System.IO.StreamWriter` defaults to using an instance of `System.Text.UTF8Encoding`
30 unless specified otherwise. This instance of `System.Text.UTF8Encoding` is constructed
31 such that the `System.Text.Encoding.GetPreamble` method returns the Unicode byte
32 order mark written in UTF-8. The preamble of the encoding is added to a stream when
33 you are not appending to an existing stream. This means any text file you create with
34 `System.IO.StreamWriter` has three byte order marks at its beginning. UTF-8 handles
35 all Unicode characters correctly and gives consistent results on localized versions of the
36 operating system.

37
38 [*Note:* By default, `System.IO.StreamWriter` is not thread safe. For a thread-safe
39 wrapper, see `System.IO.TextWriter.Synchronized`.]

1
2
3

StreamWriter(System.String, System.Boolean, System.Text.Encoding, System.Int32) Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor(string path, bool  
append, class System.Text.Encoding encoding, int32 bufferSize)
```

```
[C#]  
public StreamWriter(string path, bool append, Encoding encoding, int  
bufferSize)
```

Summary

Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the specified file on the specified path, using the specified encoding and buffer size.

Parameters

Parameter	Description
<i>path</i>	A <code>System.String</code> that specifies the complete file path to write to.
<i>append</i>	A <code>System.Boolean</code> value that determines whether data is to be appended to the file. If the file exists and <i>append</i> is <code>false</code> , the file is overwritten. If the file exists and <i>append</i> is <code>true</code> , the data is appended to the file. Otherwise, a new file is created.
<i>encoding</i>	A <code>System.Text.Encoding</code> that specifies the character encoding to use.
<i>bufferSize</i>	A <code>System.Int32</code> that specifies the buffer size.

Description

If the specified file exists, it can be either overwritten or appended to. If the file does not exist, this constructor creates a new file.

This constructor initializes the `System.IO.StreamWriter.Encoding` property using *encoding*. For additional information, see `System.IO.TextWriter.Encoding`.

[*Note:* *path* is not required to be a file stored on disk; it can be any part of a system that supports access via streams. For example, depending on the system, this class might be able to access a physical device.

For information on the valid format and characters for path strings, see `System.IO.Path`.

1
2]

3 Exceptions

Exception	Condition
System.IO.IOException	A general I/O exception occurred, such as trying to access a CD-ROM drive whose tray is open.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is a zero-length string, contains only white space, or contains one or more implementation-specific invalid characters.
System.ArgumentNullException	<i>path</i> or <i>encoding</i> is null.
System.IO.NotSupportedException	<i>path</i> is in an implementation-specific invalid format.
System.IO.PathTooLongException	The length of <i>path</i> or the absolute path information for <i>path</i> exceeds the implementation-specific maximum length.
System.ArgumentOutOfRangeException	<i>bufferSize</i> is negative.
System.Security.SecurityException	The caller does not have the required permission.
System.UnauthorizedAccessException	Access is denied. The caller does not have the required permission.

4 5 Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission for reading and writing files. See <code>System.Security.Permissions.FileIOPermissionAccess.Read</code> , <code>System.Security.Permissions.FileIOPermissionAccess.Write</code>

1

2

1 StreamWriter(System.String, 2 System.Boolean, System.Text.Encoding) 3 Constructor

```
4 [ILAsm]  
5 public rtspecialname specialname instance void .ctor(string path, bool  
6 append, class System.Text.Encoding encoding)  
  
7 [C#]  
8 public StreamWriter(string path, bool append, Encoding encoding)
```

9 Summary

10 Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the
11 specified file on the specified path, using the specified encoding and default buffer size.

12 Parameters

Parameter	Description
<i>path</i>	A <code>System.String</code> that specifies the complete file path to write to.
<i>append</i>	A <code>System.Boolean</code> value that determines whether data is to be appended to the file. If the file exists and <i>append</i> is <code>false</code> , the file is overwritten. If the file exists and <i>append</i> is <code>true</code> , the data is appended to the file. Otherwise, a new file is created.
<i>encoding</i>	A <code>System.Text.Encoding</code> that specifies the character encoding to use.

13

14 Description

15 If the specified file exists, it can be either overwritten or appended to. If the file does
16 not exist, this constructor creates a new file.

17

18 This constructor initializes the `System.IO.StreamWriter.Encoding` property using
19 *encoding*. For additional information, see `System.IO.TextWriter.Encoding`.

20

21 [*Note:* *path* is not required to be a file stored on disk; it can be any part of a system
22 that supports access via streams. For example, depending on the system, this class
23 might be able to access a physical device.

24

25 For information on the valid format and characters for path strings, see
26 `System.IO.Path`.

27

28 The default buffer size can typically be around 4 KB.

1
2]

3 Exceptions

Exception	Condition
System.IO.IOException	A general I/O exception occurred, such as trying to access a CD-ROM drive whose tray is open.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.UnauthorizedAccessException	Access is denied. The caller does not have the required permission.
System.ArgumentException	<i>path</i> is a zero-length string, contains only white space, or contains one or more implementation-specific invalid characters.
System.ArgumentNullException	<i>path</i> or <i>encoding</i> is null.
System.IO.NotSupportedException	<i>path</i> is in an implementation-specific invalid format.
System.IO.PathTooLongException	The length of <i>path</i> or the absolute path information for <i>path</i> exceeds the implementation-specific maximum length.
System.Security.SecurityException	The caller does not have the required permission.

4 5 Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission for reading and writing files. See <code>System.Security.Permissions.FileIOPermissionAccess.Read</code> , <code>System.Security.Permissions.FileIOPermissionAccess.Write</code>

6
7

StreamWriter(System.String, System.Boolean) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(string path, bool
append)

[C#]
public StreamWriter(string path, bool append)
```

Summary

Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the specified file on the specified path, using the default encoding and buffer size.

Parameters

Parameter	Description
<i>path</i>	A <code>System.String</code> that specifies the complete file path to write to.
<i>append</i>	A <code>System.Boolean</code> value that determines whether data is to be appended to the file. If the file exists and <i>append</i> is <code>false</code> , the file is overwritten. If the file exists and <i>append</i> is <code>true</code> , the data is appended to the file. Otherwise, a new file is created.

Description

This constructor initializes the `System.IO.StreamWriter.Encoding` property to `System.Text.UTF8Encoding` whose `System.Text.Encoding.GetPreamble` method returns an empty byte array. For additional information, see `System.IO.TextWriter.Encoding`.

If the specified file exists, it can be either overwritten or appended to. If the file does not exist, this constructor creates a new file.

[*Note:* *path* is not required to be a file stored on disk; it can be any part of a system that supports access via streams. For example, depending on the system, this class might be able to access a physical device.

For information on the valid format and characters for path strings, see `System.IO.Path`.

The default buffer size can typically be around 4 KB.

]

1 Exceptions

Exception	Condition
System.IO.IOException	A general I/O exception occurs, such as trying to access a CD-ROM drive whose tray is open
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.UnauthorizedAccessException	Access to <i>path</i> is denied. The caller does not have the required permission.
System.ArgumentException	<i>path</i> is a zero-length string, contains only white space, or contains one or more implementation-specific invalid characters.
System.IO.NotSupportedException	<i>path</i> is in an implementation-specific invalid format.
System.IO.PathTooLongException	The length of <i>path</i> or the absolute path information for <i>path</i> exceeds the implementation-specific maximum length.
System.ArgumentNullException	<i>path</i> is null.
System.Security.SecurityException	The caller does not have the required permission.

2

3 Permissions

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission for reading and writing files. See <code>System.Security.Permissions.FileIOPermissionAccess.Read</code> , <code>System.Security.Permissions.FileIOPermissionAccess.Write</code>

4

5

1 StreamWriter(System.IO.Stream)

2 Constructor

```
3 [ILAsm]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.IO.Stream stream)  
  
6 [C#]  
7 public StreamWriter(Stream stream)
```

8 Summary

9 Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the
10 specified stream, using the default encoding and buffer size.

11 Parameters

Parameter	Description
<i>stream</i>	The <code>System.IO.Stream</code> to write to.

12 Description

14 This constructor initializes the `System.IO.StreamWriter.Encoding` property to a
15 `System.Text.UTF8Encoding` whose `System.Text.Encoding.GetPreamble` method
16 returns an empty byte array. For additional information, see
17 `System.IO.TextWriter.Encoding`. The `System.IO.StreamWriter.BaseStream` property
18 is initialized using *stream*.

19
20 [Note: The default buffer size can typically be around 4 KB.]
21
22

23 Exceptions

Exception	Condition
System.ArgumentException	<i>stream</i> does not support writing.
System.ArgumentNullException	<i>stream</i> is null.

24

25

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

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

[C#]
public StreamWriter(Stream stream, Encoding encoding)
```

Summary

Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the specified stream, using the specified encoding and the default buffer size.

Parameters

Parameter	Description
<i>stream</i>	The <code>System.IO.Stream</code> to write to.
<i>encoding</i>	A <code>System.Text.Encoding</code> that specifies the character encoding to use.

Description

This constructor initializes the `System.IO.StreamWriter.Encoding` property using *encoding*, and the `System.IO.StreamWriter.BaseStream` property using *stream*. For additional information, see `System.IO.TextWriter.Encoding`.

[*Note:* The default buffer size can typically be around 4 KB.]

Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>stream</i> or <i>encoding</i> is null.
<code>System.ArgumentException</code>	<i>stream</i> does not support writing.

1 StreamWriter(System.IO.Stream, 2 System.Text.Encoding, System.Int32) 3 Constructor

```
4 [ILAsm]  
5 public rtspecialname specialname instance void .ctor(class  
6 System.IO.Stream stream, class System.Text.Encoding encoding, int32  
7 bufferSize)  
  
8 [C#]  
9 public StreamWriter(Stream stream, Encoding encoding, int bufferSize)
```

10 Summary

11 Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the
12 specified stream, using the specified encoding and buffer size.

13 Parameters

Parameter	Description
<i>stream</i>	The <code>System.IO.Stream</code> to write to.
<i>encoding</i>	A <code>System.Text.Encoding</code> that specifies the character encoding to use.
<i>bufferSize</i>	A <code>System.Int32</code> that specifies the buffer size.

14

15 Description

16 This constructor initializes the `System.IO.StreamWriter.Encoding` property using
17 *encoding*, and the `System.IO.StreamWriter.BaseStream` property using *stream*. For
18 additional information, see `System.IO.TextWriter.Encoding`.

19 Exceptions

Exception	Condition
System.ArgumentNullException	<i>stream</i> or <i>encoding</i> is null.
System.ArgumentOutOfRangeException	<i>bufferSize</i> is negative.
System.ArgumentException	<i>stream</i> does not support writing.

20

21

StreamWriter(System.String) Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor(string path)  
  
[C#]  
public StreamWriter(string path)
```

Summary

Constructs and initializes a new instance of the `System.IO.StreamWriter` class for the specified file on the specified path, using the default encoding and buffer size.

Parameters

Parameter	Description
<i>path</i>	A <code>System.String</code> that specifies the complete file path to write to.

Description

This constructor initializes the `System.IO.StreamWriter.Encoding` property to a `System.Text.UTF8Encoding` whose `System.Text.Encoding.GetPreamble` method returns an empty byte array. For additional information, see `System.IO.TextWriter.Encoding`.

[*Note:* *path* is not required to be a file stored on disk; it can be any part of a system that supports access via streams. For example, depending on the system, this class might be able to access a physical device.

For information on the valid format and characters for path strings, see `System.IO.Path`.

The default buffer size can typically be around 4 KB.

]

Exceptions

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.

System.UnauthorizedAccessException	Access to <i>path</i> is denied.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> is null.
System.IO.PathTooLongException	The length of <i>path</i> or the absolute path information for <i>path</i> exceeds the implementation-specific maximum length.
System.Security.SecurityException	The caller does not have the required permission.

1

2 **Permissions**

Permission	Description
System.Security.Permissions.FileIOPermission	Requires permission for reading and writing files. See <code>System.Security.Permissions.FileIOPermissionAccess.Read</code> , <code>System.Security.Permissions.FileIOPermissionAccess.Write</code>

3

4

1 StreamWriter.Close() Method

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

6 Summary

7 Closes the current `System.IO.StreamWriter` and the underlying stream.

8 Description

9 This method calls `System.IO.StreamWriter.Flush`, writing buffered data to the
10 underlying stream. Following a call to `System.IO.StreamWriter.Close`, any operations
11 on the current instance might raise exceptions.

12
13 [*Note:* This version of `System.IO.StreamWriter.Close` is equivalent to
14 `System.IO.StreamWriter.Dispose(true)`.

15
16 This method overrides `System.IO.Stream.Close`.

17
18]

19

1 StreamWriter.Dispose(System.Boolean)

2 Method

```
3 [ILAsm]  
4 .method family hidebysig virtual void Dispose(bool disposing)  
5 [C#]  
6 protected override void Dispose(bool disposing)
```

7 Summary

8 Releases the unmanaged resources used by the `System.IO.StreamWriter` and
9 optionally releases the managed resources.

10 Parameters

Parameter	Description
<i>disposing</i>	true to release both managed and unmanaged resources; false to release only unmanaged resources.

11 Description

12

13 When the *disposing* parameter is true, this method releases all resources held by any
14 managed objects that this `System.IO.StreamWriter` references. This method invokes
15 the `Dispose()` method of each referenced object.

16
17 [Note: `System.IO.StreamWriter.Dispose` can be called multiple times by other objects.
18 When overriding `System.IO.StreamWriter.Dispose(System.Boolean)`, be careful not
19 to reference objects that have been previously disposed in an earlier call to
20 `System.IO.StreamWriter.Dispose`.

21
22 This method calls the dispose method of the base class,
23 `System.IO.TextWriter.Dispose(disposing)`.

24]
25

26

1 StreamWriter.Finalize() Method

```
2 [ILAsm]  
3 .method family hidebysig virtual void Finalize()  
4 [C#]  
5 ~StreamWriter()
```

6 Summary

7 Releases resources held by the current instance.

8 Description

9 *[Note:* Application code does not call this method; it is automatically invoked by during
10 garbage collection unless finalization by the garbage collector has been disabled. For
11 more information, see `System.GC.SuppressFinalize`, and `System.Object.Finalize`.
12
13 This method overrides `System.Object.Finalize`.
14
15]

16

1 StreamWriter.Flush() Method

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

6 Summary

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

9 Description

10 *[Note: This method overrides System.IO.TextWriter.Flush.]*
11
12

13 Exceptions

Exception	Condition
System.ObjectDisposedException	The current writer is closed.
System.IO.IOException	An I/O error occurred.

14

15

1 **StreamWriter.Write(System.Char) Method**

```
2 [ILAsm]
3 .method public hidebysig virtual void Write(valuetype System.Char value)
4 [C#]
5 public override void Write(char value)
```

6 **Summary**

7 Writes a character to the stream.

8 **Parameters**

Parameter	Description
<i>value</i>	The System.Char to write to the underlying stream.

9
10 **Description**

11 The specified character is written to the underlying stream unless the end of the stream
12 is reached prematurely.

13
14 If System.IO.StreamWriter.AutoFlush is true, System.IO.StreamWriter.Flush is
15 invoked automatically.

16
17 [*Note:* This method overrides System.IO.TextWriter.Write.]
18
19

20 **Exceptions**

Exception	Condition
System.NotSupportedException	System.IO.StreamWriter.AutoFlush is true or the System.IO.StreamWriter buffer is full, and the contents of the buffer cannot be written to the underlying fixed size stream because the System.IO.StreamWriter is at the end the stream.
System.ObjectDisposedException	The current writer is closed.
System.IO.IOException	An I/O error occurred.

21
22

1 StreamWriter.Write(System.Char[]) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(char[] buffer)  
4 [C#]  
5 public override void Write(char[] buffer)
```

6 Summary

7 Writes a character array to the underlying stream.

8 Parameters

Parameter	Description
<i>buffer</i>	A System.Char array containing the data to write. If <i>buffer</i> is null, nothing is written.

10 Description

11 The specified characters are written to the underlying stream unless the end of the
12 stream is reached prematurely.

13
14 If System.IO.StreamWriter.AutoFlush is true, System.IO.StreamWriter.Flush is
15 invoked automatically.

16
17 [*Note:* This method overrides System.IO.TextWriter.Write.]
18
19

20 Exceptions

Exception	Condition
System.NotSupportedException	System.IO.StreamWriter.AutoFlush is true or the System.IO.StreamWriter buffer is full, and the contents of the buffer cannot be written to the underlying fixed size stream because the System.IO.StreamWriter is at the end the stream.
System.ObjectDisposedException	The current writer is closed.
System.IO.IOException	An I/O error occurred.

1 StreamWriter.Write(System.Char[], 2 System.Int32, System.Int32) Method

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

8 Summary

9 Writes a sub-array of characters to the underlying stream.

10 Parameters

Parameter	Description
<i>buffer</i>	A System.Char array containing the data to write.
<i>index</i>	A System.Int32 that specifies the index into <i>buffer</i> at which to begin writing.
<i>count</i>	A System.Int32 that specifies the number of characters to read from <i>buffer</i> .

11

12 Description

13 The specified characters are written to the underlying stream unless the end of the
14 stream is reached prematurely.

15
16 If System.IO.StreamWriter.AutoFlush is true, System.IO.StreamWriter.Flush is
17 invoked automatically.

18
19 [Note: This method overrides System.IO.TextWriter.Write.]
20
21

22 Exceptions

Exception	Condition
System.ArgumentException	<i>buffer.Length - index < count.</i>
System.ArgumentNullException	<i>buffer</i> is null.
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is negative.

System.NotSupportedException	System.IO.StreamWriter.AutoFlush is true or the System.IO.StreamWriter buffer is full, and the contents of the buffer cannot be written to the underlying fixed size stream because the System.IO.StreamWriter is at the end the stream.
System.ObjectDisposedException	The current writer is closed.
System.IO.IOException	An I/O error occurred.

1

2

1 StreamWriter.Write(System.String) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(string value)  
4 [C#]  
5 public override void Write(string value)
```

6 Summary

7 Writes a string to the stream.

8 Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to write to the stream. If <i>value</i> is null, nothing is written.

9 Description

11 The specified `System.String` is written to the underlying stream unless the end of the
12 stream is reached prematurely.

13
14 If `System.IO.StreamWriter.AutoFlush` is true, `System.IO.StreamWriter.Flush` is
15 invoked automatically.

16
17 [*Note:* This method overrides `System.IO.TextWriter.Write`.]
18
19

20 Exceptions

Exception	Condition
System.NotSupportedException	<code>System.IO.StreamWriter.AutoFlush</code> is true or the <code>System.IO.StreamWriter</code> buffer is full, and the contents of the buffer cannot be written to the underlying fixed size stream because the <code>System.IO.StreamWriter</code> is at the end the stream.
System.ObjectDisposedException	The current writer is closed.
System.IO.IOException	An I/O error occurred.

21

22

1 StreamWriter.AutoFlush Property

```
2 [ILAsm]  
3 .property bool AutoFlush { public hidebysig virtual specialname bool  
4 get_AutoFlush() public hidebysig virtual specialname void  
5 set_AutoFlush(bool value) }  
6 [C#]  
7 public virtual bool AutoFlush { get; set; }
```

8 Summary

9 Gets or sets a System.Boolean value indicating whether the current
10 System.IO.StreamWriter will flush its buffer to the underlying stream after every call
11 to System.IO.StreamWriter.Write.

12 Property Value

13 true to force System.IO.StreamWriter to flush its buffer; otherwise, false.

14 Description

15 The System.IO.StreamWriter will do a limited amount of buffering, both internally and
16 potentially in the encoder from the encoding you passed in. If
17 System.IO.StreamWriter.AutoFlush is set to false, the data will be flushed into the
18 underlying stream only when the buffer is full, or when
19 System.IO.StreamWriter.Dispose(true) or System.IO.StreamWriter.Close is called.
20

21 Setting System.IO.StreamWriter.AutoFlush to true forces System.IO.StreamWriter
22 to flush the buffered data out of the encoder and call System.IO.StreamWriter.Flush
23 on the stream every time System.IO.StreamWriter.Write is called.

24 Behaviors

25 As described above.

26

27

1 StreamWriter.BaseStream Property

```
2 [ILAsm]  
3 .property class System.IO.Stream BaseStream { public hidebysig virtual  
4 specialname class System.IO.Stream get_BaseStream() }  
5 [C#]  
6 public virtual Stream BaseStream { get; }
```

7 Summary

8 Gets the underlying stream.

9 Property Value

10 The System.IO.Stream the current StreamWriter instance is writing to.

11 Behaviors

12 As described above.

13

14

1 StreamWriter.Encoding Property

```
2 [ILAsm]  
3 .property class System.Text.Encoding Encoding { public hidebysig virtual  
4 specialname class System.Text.Encoding get_Encoding() }  
5 [C#]  
6 public override Encoding Encoding { get; }
```

7 Summary

8 Gets the `System.Text.Encoding` in which the output is written.

9 Property Value

10 The `System.Text.Encoding` specified in the constructor for the current instance, or
11 `System.Text.UTF8Encoding` if an encoding was not specified.

12 Description

13 [*Note:* This property overrides the `System.IO.TextWriter.Encoding` property.]
14
15

16 Behaviors

17 As described above.
18

19 Usage

20 This property is required in some XML scenarios where a header must be written
21 containing the encoding used by the `System.IO.StreamWriter`. This allows XML code to
22 consume an arbitrary `System.IO.StreamWriter` and generate a correct XML header.
23
24