

# 1 System.IO.TextWriter Class

```
2 [ILAsm]  
3 .class public abstract serializable TextWriter extends  
4 System.MarshalByRefObject implements System.IDisposable  
  
5 [C#]  
6 public abstract class TextWriter: MarshalByRefObject, IDisposable
```

## 7 Assembly Info:

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

## 13 Implements:

- 14 • **System.IDisposable**

## 15 Summary

16 Represents a writer that can write a sequential series of characters.

## 17 Inherits From: System.MarshalByRefObject

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.

## 23 24 Description

25 System.IO.TextWriter is designed for character output, whereas the  
26 System.IO.Stream class is designed for byte input and output.

27  
28 [*Note:* By default, a System.IO.TextWriter is not thread safe. See  
29 System.IO.TextWriter.Synchronized for a thread-safe wrapper.]  
30  
31

32

# 1 TextWriter(System.IFormatProvider)

## 2 Constructor

```
3 [ILAsm]  
4 family rtspecialname specialname instance void .ctor(class  
5 System.IFormatProvider formatProvider)  
  
6 [C#]  
7 protected TextWriter(IFormatProvider formatProvider)
```

### 8 Summary

9 Constructs and initializes a new instance of the `System.IO.TextWriter` class with the  
10 specified format provider.

### 11 Parameters

Parameter	Description
<i>formatProvider</i>	A <code>System.IFormatProvider</code> object that supplies a formatting object.

12

13

# 1 TextWriter() Constructor

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

## 6 Summary

7 Constructs and initializes a new instance of the System.IO.TextWriter class.

8

# 1 TextWriter.Null Field

```
2 [ILAsm]  
3 .field public static initOnly class System.IO.TextWriter Null  
4 [C#]  
5 public static readonly TextWriter Null
```

## 6 Summary

7 Provides a `System.IO.TextWriter` object without a backing store.

## 8 Description

9 Use `System.IO.TextWriter.Null` to redirect output to a stream that will not consume  
10 any operating system resources.

11  
12 When a `System.IO.TextWriter.Write` method is invoked on  
13 `System.IO.TextWriter.Null`, the call simply returns, and no data is actually written to  
14 any backing store.

15

# 1 TextWriter.Close() Method

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

## 6 Summary

7 Closes the current `System.IO.TextWriter` instance and releases any system resources  
8 associated with it.

## 9 Description

10 [*Note:* After a call to `System.IO.TextReader.Close`, any operation on the current  
11 instance might throw an exception.

12 This version of `System.IO.TextReader.Close` is equivalent to  
13 `System.IO.TextReader.Dispose(true)`.

14 ]  
15 ]  
16 ]

## 17 Behaviors

18 As described above.

19

20

# 1 TextWriter.Dispose(System.Boolean) Method

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

## 6 Summary

7 Releases the unmanaged resources used by the `System.IO.TextWriter` and optionally  
8 releases the managed resources.

## 9 Parameters

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

10

## 11 Description

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

15

16 [Note: `System.IO.FileStream.Dispose` can be called multiple times by other objects.  
17 When overriding `System.IO.FileStream.Dispose(System.Boolean)`, be careful not to  
18 reference objects that have been previously disposed in an earlier call to  
19 `System.IO.FileStream.Dispose`.]  
20  
21

20

21

22

# 1 TextWriter.Flush() Method

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

## 6 Summary

7 Writes any buffered data to the underlying device and clears all buffers for the current  
8 System.IO.TextWriter instance.

## 9 Behaviors

10 As described above.

11

## 12 Default

13 This method is not implemented in this abstract class.

14

15



# 1 TextWriter.System.IDisposable.Dispose() 2 Method

```
3 [ILAsm]  
4 .method private final hidebysig virtual void System.IDisposable.Dispose()  
5 [C#]  
6 void IDisposable.Dispose()
```

## 7 Summary

8 Implemented to support the System.IDisposable interface. [Note: For more  
9 information, see System.IDisposable.Dispose.]

10

# 1 TextWriter.Write(System.String, 2 System.Object[]) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void Write(string format, object[] arg)  
5  
6 [C#]  
7 public virtual void Write(string format, params object[] arg)
```

## 7 Summary

8 Writes a formatted string to the text stream.

## 9 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg</i>	The array of objects referenced from the format string.

10

## 11 Description

12 This version of System.IO.TextWriter.Write is equivalent to  
13 System.IO.TextWriter.Write(System.String.Format (*format*, *arg*)).

14

15 [*Note*: If a specified object is not referenced in the format string, it is ignored.

16

17 This method does not search the specified System.String for individual newline  
18 characters (hexadecimal 0x000a) and replace them with  
19 System.IO.TextWriter.NewLine.

20

21 ]

## 22 Behaviors

23 As described above.

24

## 25 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.

	-or-  <i>arg</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	The format specification in <i>format</i> is invalid.  -or-  The number indicating an argument to be formatted is less than zero, or greater than or equal to <i>arg.Length</i> .

1

2

# 1 TextWriter.Write(System.String, 2 System.Object, System.Object, 3 System.Object) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual void Write(string format, object arg0,  
6 object arg1, object arg2)  
  
7 [C#]  
8 public virtual void Write(string format, object arg0, object arg1, object  
9 arg2)
```

## 10 Summary

11 Writes out a formatted string to the text stream.

## 12 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.
<i>arg2</i>	The third object referenced in the format string.

## 13 14 Description

15 This version of System.IO.TextWriter.Write is equivalent to  
16 System.IO.TextWriter.Write(System.String.Format (*format*, *arg0*, *arg1*, *arg2*)).

17  
18 [Note: If a specified object is not referenced in the format string, it is ignored.

19  
20 This method does not search the specified System.String for individual newline  
21 characters (hexadecimal 0x000a) and replace them with  
22 System.IO.TextWriter.NewLine.

23  
24 ]

## 25 Behaviors

26 As described above.

27

## 1 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	The format specification in <i>format</i> is invalid.  -or-  The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (3).

2

3

# 1 TextWriter.Write(System.String, 2 System.Object, System.Object) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void Write(string format, object arg0,  
5 object arg1)  
  
6 [C#]  
7 public virtual void Write(string format, object arg0, object arg1)
```

## 8 Summary

9 Writes out a formatted string to the text stream.

## 10 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.

11

## 12 Description

13 This version of System.IO.TextWriter.Write is equivalent to  
14 System.IO.TextWriter.Write(System.String.Format (*format*, *arg0*, *arg1*)).

15  
16 [*Note:* If a specified object is not referenced in the format string, it is ignored.

17  
18 This method does not search the specified System.String for individual newline  
19 characters (hexadecimal 0x000a) and replace them with  
20 System.IO.TextWriter.NewLine.

21  
22 ]

## 23 Behaviors

24 As described above.

25

## 26 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	<p>The format specification in <i>format</i> is invalid.</p> <p>-or-</p> <p>The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (2).</p>

1

2

# 1 TextWriter.Write(System.String, 2 System.Object) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void Write(string format, object arg0)  
5 [C#]  
6 public virtual void Write(string format, object arg0)
```

## 7 Summary

8 Writes out the format string to the text stream.

## 9 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	An object referenced in the format string.

10

## 11 Description

12 This version of System.IO.TextWriter.Write is equivalent to  
13 System.IO.TextWriter.Write(System.String.Format (*format*, *arg0*)).

14  
15 [*Note*: If the specified object is not referenced in the format string, it is ignored.

16  
17 This method does not search the specified System.String for individual newline  
18 characters (hexadecimal 0x000a) and replace them with  
19 System.IO.TextWriter.NewLine.

20  
21 ]

## 22 Behaviors

23 As described above.

24

## 25 Exceptions

Exception	Condition
System.ArgumentNullException	<i>format</i> is null.

<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	The format specification in <i>format</i> is invalid.  -or-  The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (1).

1

2

# 1 TextWriter.Write(System.Object) Method

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

## 6 Summary

7 Writes the text representation of a specified object to the text stream.

## 8 Parameters

Parameter	Description
<i>value</i>	The object to write. If <i>value</i> is null, nothing is written.

## 9 10 Description

11 This version of `System.IO.TextWriter.Write` is equivalent to  
12 `System.IO.TextWriter.Write( value.ToString())`.

## 13 Behaviors

14 As described above.

15

## 16 Default

17 The default implementation uses the `System.IO.TextWriter.Write(System.String)`  
18 method.

## 19 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

20

21

# 1 TextWriter.Write(System.String) Method

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

## 6 Summary

7 Writes a specified `System.String` to the text stream.

## 8 Parameters

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

## 9 10 Description

11 This version of `System.IO.TextWriter.Write` is equivalent to  
12 `System.IO.TextWriter.Write(value.ToArray())`.

13  
14 [*Note:* This method does not search the specified `System.String` for individual newline  
15 characters (hexadecimal 0x000a) and replace them with  
16 `System.IO.TextWriter.NewLine`.]  
17  
18

## 19 Behaviors

20 As described above.

21

## 22 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

23

24

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## TextWriter.Write(System.Decimal) Method

```
[ILAsm]  
.method public hidebysig virtual void Write(decimal value)  
  
[C#]  
public virtual void Write(decimal value)
```

### Summary

Writes the text representation of a specified System.Decimal to the text stream.

### Parameters

Parameter	Description
<i>value</i>	The System.Decimal to write.

### Description

This version of System.IO.TextWriter.Write is equivalent to System.IO.TextWriter.Write( *value*.ToString ()).

### Behaviors

As described above.

### Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 **TextWriter.Write(System.Double) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual void Write(float64 value)  
  
6 [C#]  
7 public virtual void Write(double value)
```

### 8 **Summary**

9 Writes the text representation of a specified System.Double to the text stream.

### 10 **Parameters**

Parameter	Description
<i>value</i>	The System.Double to write.

### 12 **Description**

13 This version of System.IO.TextWriter.Write is equivalent to  
14 System.IO.TextWriter.Write( *value*.ToString()).

### 15 **Behaviors**

16 As described above.

### 18 **Exceptions**

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

## TextWriter.Write(System.Single) Method

```
[ILAsm]
.method public hidebysig virtual void Write(float32 value)

[C#]
public virtual void Write(float value)
```

### Summary

Writes the text representation of a specified `System.Single` to the text stream.

### Parameters

Parameter	Description
<i>value</i>	The <code>System.Single</code> to write.

### Description

This version of `System.IO.TextWriter.Write` is equivalent to `System.IO.TextWriter.Write( value.ToString())`.

### Behaviors

As described above.

### Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

# 1 TextWriter.Write(System.UInt64) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(unsigned int64 value)  
4 [C#]  
5 public virtual void Write(ulong value)
```

## 6 Summary

7 Writes the text representation of a specified System.UInt64 to the text stream.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.UInt64 to write.

11

## 12 Description

13 This member is not CLS-compliant. For a CLS-compliant alternative, use  
14 System.IO.TextWriter.Write(System.Decimal).

15  
16 This version of System.IO.TextWriter.Write is equivalent to  
17 System.IO.TextWriter.Write( *value*.ToString () ).

## 18 Behaviors

19 As described above.

20

## 21 Exceptions

Exception	Condition
System.IO.IOException	An I/O error occurred.

22

23

# 1 TextWriter.Write(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(int64 value)  
4 [C#]  
5 public virtual void Write(long value)
```

## 6 Summary

7 Writes the text representation of a specified `System.Int64` to the text stream.

## 8 Parameters

Parameter	Description
<i>value</i>	The <code>System.Int64</code> to write.

## 9 10 Description

11 This version of `System.IO.TextWriter.Write` is equivalent to  
12 `System.IO.TextWriter.Write( value.ToString () )`.

## 13 Behaviors

14 As described above.

15

## 16 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

17

18

# 1 TextWriter.Write(System.UInt32) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(unsigned int32 value)  
4 [C#]  
5 public virtual void Write(uint value)
```

## 6 Summary

7 Writes the text representation of a System.UInt32 to the text stream.

## 8 Type Attributes:

- 9 • CLSCompliantAttribute(false)

## 10 Parameters

Parameter	Description
<i>value</i>	The System.UInt32 to write.

11

## 12 Description

13 This member is not CLS-compliant. For a CLS-compliant alternative, use  
14 System.IO.TextWriter.Write(System.Int64).

15  
16 This version of System.IO.TextWriter.Write is equivalent to  
17 System.IO.TextWriter.Write( *value*.ToString ()).

## 18 Behaviors

19 As described above.

20

## 21 Exceptions

Exception	Condition
System.IO.IOException	An I/O error occurred.

22

23

# 1 TextWriter.Write(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Write(int32 value)  
4 [C#]  
5 public virtual void Write(int value)
```

## 6 Summary

7 Writes the text representation of a specified `System.Int32` to the text stream.

## 8 Parameters

Parameter	Description
<i>value</i>	The <code>System.Int32</code> to write.

## 9 10 Description

11 This version of `System.IO.TextWriter.Write` is equivalent to  
12 `System.IO.TextWriter.Write( value.ToString ()`).

## 13 Behaviors

14 As described above.

15

## 16 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

17

18

# 1 TextWriter.Write(System.Boolean) Method

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

## 6 Summary

7 Writes the text representation of a specified System.Boolean value to the text stream.

## 8 Parameters

Parameter	Description
<i>value</i>	The System.Boolean value to write.

## 9 Description

11 This version of System.IO.TextWriter.Write is equivalent to  
12 System.IO.TextWriter.Write( *value*.ToString ()).

## 13 Behaviors

14 As described above.

## 16 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

17  
18

# 1 `TextWriter.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 virtual void Write(char[] buffer, int index, int count)
```

## 8 **Summary**

9 Writes a subarray of characters to the text stream.

## 10 **Parameters**

Parameter	Description
<i>buffer</i>	The character array from which characters are read.
<i>index</i>	Starting index in the buffer.
<i>count</i>	The number of characters to write.

11

## 12 **Description**

13 [Note: This method does not search the specified array for individual newline characters  
14 (hexadecimal 0x000a) and replace them with `System.IO.TextWriter.NewLine`.]  
15  
16

## 17 **Behaviors**

18 As described above.  
19

## 20 **Default**

21 The default implementation uses the `System.IO.TextWriter.Write(char[])` method.

## 22 **Exceptions**

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

<b>System.ArgumentNullException</b>	<i>buffer</i> is null.
<b>System.ArgumentException</b>	$(index + count) > buffer.Length$ .
<b>System.ArgumentOutOfRangeException</b>	$index < 0$ or $count < 0$ .
<b>System.IO.IOException</b>	An I/O error occurred.

1

2

# 1 TextWriter.Write(System.Char[]) Method

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

## 6 Summary

7 Writes a character array to the text stream.

## 8 Parameters

Parameter	Description
<i>buffer</i>	The System.Char array to write to the text stream. If <i>buffer</i> is null, nothing is written.

9

## 10 Description

11 This version of System.IO.TextWriter.Write is equivalent to  
12 System.IO.TextWriter.Write(*buffer*, 0, *buffer*.Length ).

13

14 [*Note:* This method does not search the specified array for individual newline characters  
15 (hexadecimal 0x000a) and replace them with System.IO.TextWriter.NewLine.]

16

17

## 18 Behaviors

19 As described above.

20

## 21 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

22

23

# 1 TextWriter.Write(System.Char) Method

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

## 6 Summary

7 Writes a character to the text stream.

## 8 Parameters

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

9

## 10 Behaviors

11 As described above.

12

## 13 Default

14 This method is not implemented in this abstract class.

15

## 16 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

17

18

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

## 3 `TextWriter.WriteLine(System.Decimal)` 4 Method

```
5 [ILAsm]  
6 .method public hidebysig virtual void WriteLine(decimal value)  
7 [C#]  
8 public virtual void WriteLine(decimal value)
```

### 9 Summary

10 Writes the text representation of a specified `System.Decimal` value followed by a line  
11 terminator to the text stream.

### 12 Parameters

Parameter	Description
<i>value</i>	The <code>System.Decimal</code> value to write.

### 14 Description

15 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
16 `System.IO.TextWriter.WriteLine (value.ToString())`.

### 17 Behaviors

18 As described above.

### 20 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

# 1 TextWriter.WriteLine(System.Int64) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void WriteLine(int64 value)  
4 [C#]  
5 public virtual void WriteLine(long value)
```

## 6 Summary

7 Writes the text representation of a specified `System.Int64` value followed by a line  
8 terminator to the text stream.

## 9 Parameters

Parameter	Description
<i>value</i>	The <code>System.Int64</code> value to write.

10

## 11 Description

12 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
13 `System.IO.TextWriter.WriteLine( value.ToString () )`.

## 14 Behaviors

15 As described above.

16

## 17 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

18

19

# 1 StreamWriter.WriteLine(System.UInt32)

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(unsigned int32 value)  
5 [C#]  
6 public virtual void WriteLine(uint value)
```

### 7 Summary

8 Writes the text representation of a specified `System.UInt32` value followed by a line  
9 terminator to the text stream.

### 10 Type Attributes:

- 11 • `CLSCompliantAttribute(false)`

### 12 Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt32</code> value to write.

### 13 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 `System.IO.TextWriter.WriteLine(System.Int64)`.

17 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
18 `System.IO.TextWriter.WriteLine( value.ToString() )`.

### 20 Behaviors

21 As described above.

### 23 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

24  
25

# 1 TextWriter.WriteLine(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void WriteLine(int32 value)  
4 [C#]  
5 public virtual void WriteLine(int value)
```

## 6 Summary

7 Writes the text representation of a specified `System.Int32` value followed by a line  
8 terminator to the text stream.

## 9 Parameters

Parameter	Description
<i>value</i>	The <code>System.Int32</code> value to write.

10

## 11 Description

12 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
13 `System.IO.TextWriter.WriteLine(value.ToString())`.

## 14 Behaviors

15 As described above.

16

## 17 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

18

19

# 1 TextWriter.WriteLine(System.Boolean)

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(bool value)  
5 [C#]  
6 public virtual void WriteLine(bool value)
```

### 7 Summary

8 Writes the text representation of a `System.Boolean` value followed by a line terminator  
9 to the text stream.

### 10 Parameters

Parameter	Description
<i>value</i>	The <code>System.Boolean</code> value to write.

### 11 Description

12 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
13 `System.IO.TextWriter.WriteLine( value.ToString () )`.

### 15 Behaviors

16 As described above.

### 18 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

19  
20

# 1 `TextWriter.WriteLine(System.Char[],` 2 `System.Int32, System.Int32) Method`

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

## 8 **Summary**

9 Writes a sub-array of characters from a specified character array, followed by a line  
10 terminator to the text stream.

## 11 **Parameters**

Parameter	Description
<i>buffer</i>	The <code>System.Char</code> array from which characters are read.
<i>index</i>	A <code>System.Int32</code> that specifies the index in <i>buffer</i> at which to begin reading.
<i>count</i>	A <code>System.Int32</code> that specifies the maximum number of characters to write.

12

## 13 **Description**

14 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
15 `System.IO.TextWriter.Write( value, index, count )` followed by  
16 `System.IO.TextWriter.WriteLine()`.

17

18 [*Note:* This method does not search the specified array for individual newline characters  
19 (hexadecimal 0x000a) and replace them with `System.IO.TextWriter.NewLine`.]  
20  
21

## 22 **Behaviors**

23 As described above.

24

## 25 **Exceptions**

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

<b>System.ArgumentNullException</b>	<i>buffer</i> is null.
<b>System.ArgumentException</b>	$(index + count) > buffer.Length$ .
<b>System.ArgumentOutOfRangeException</b>	$index < 0$ -or- $count < 0$ .
<b>System.IO.IOException</b>	An I/O error occurred.

1

2

# 1 TextWriter.WriteLine(System.Char[]) Method

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

## 6 Summary

7 Writes a specified character array followed by a line terminator to the text stream.

## 8 Parameters

Parameter	Description
<i>buffer</i>	The System.Char array from which data is read. If <i>buffer</i> is null, only the line terminator is written.

## 9 Description

11 This version of System.IO.TextWriter.WriteLine is equivalent to  
12 System.IO.TextWriter.Write( *value* ) followed by  
13 System.IO.TextWriter.WriteLine().

14  
15 [Note: This method does not search the specified array for individual newline characters  
16 (hexadecimal 0x000a) and replace them with System.IO.TextWriter.NewLine.]  
17  
18

## 19 Behaviors

20 As described above.

## 22 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

# 1 TextWriter.WriteLine(System.Char) Method

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

## 7 Summary

8 Writes a specified character followed by a line terminator to the text stream.

## 9 Parameters

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

10

## 11 Description

12 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
13 `System.IO.TextWriter.Write( value )` followed by  
14 `System.IO.TextWriter.WriteLine()`.

## 15 Behaviors

16 As described above.

17

## 18 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

19

20

# 1 TextWriter.WriteLine() Method

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

## 6 Summary

7 Writes a line terminator to the text stream.

## 8 Description

9 This version of System.IO.TextWriter.WriteLine is equivalent to  
10 System.IO.TextWriter.Write(System.IO.TextWriter.NewLine.ToCharArray()).

## 11 Behaviors

12 As described above.

13

## 14 Exceptions

Exception	Condition
System.IO.IOException	An I/O error occurred.

15

16

# 1 StreamWriter.WriteLine(System.UInt64)

## 2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(unsigned int64 value)  
5 [C#]  
6 public virtual void WriteLine(ulong value)
```

### 7 Summary

8 Writes the text representation of a specified `System.UInt64` value followed by a line  
9 terminator to the text stream.

### 10 Type Attributes:

- 11 • `CLSCompliantAttribute(false)`

### 12 Parameters

Parameter	Description
<i>value</i>	The <code>System.UInt64</code> value to write.

### 13 Description

15 This member is not CLS-compliant. For a CLS-compliant alternative, use  
16 `System.IO.TextWriter.WriteLine(System.Decimal)`.

17 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
18 `System.IO.TextWriter.WriteLine( value.ToString() )`.

### 20 Behaviors

21 As described above.

### 23 Exceptions

Exception	Condition
<code>System.IO.IOException</code>	An I/O error occurred.

24  
25

# 1 TextWriter.WriteLine(System.String) Method

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

## 6 Summary

7 Writes a specified `System.String` followed by a line terminator to the text stream.

## 8 Parameters

Parameter	Description
<i>value</i>	The <code>System.String</code> to write. If <i>value</i> is null, only the line terminator is written.

9

## 10 Description

11 This version of `System.IO.TextWriter.Write` is equivalent to  
12 `System.IO.TextWriter.Write( value.ToArray() )`.

13

14 The line terminator string is defined by the `System.IO.TextWriter.NewLine` property.

15

16 [*Note:* This method does not search the specified `System.String` for individual newline  
17 characters (hexadecimal 0x000a) and replace them with  
18 `System.IO.TextWriter.NewLine`.]

19

20

## 21 Behaviors

22 As described above.

23

## 24 Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

25

26

# 1 TextWriter.WriteLine(System.Object) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void WriteLine(object value)  
4 [C#]  
5 public virtual void WriteLine(object value)
```

## 6 Summary

7 Writes the text representation of a specified object followed by a line terminator to the  
8 text stream.

## 9 Parameters

Parameter	Description
<i>value</i>	The object to write. If <i>value</i> is null, only the line terminator is written.

10

## 11 Description

12 This version of System.IO.TextWriter.WriteLine is equivalent to  
13 System.IO.TextWriter.WriteLine( *value*.ToString() ).

## 14 Behaviors

15 As described above.

16

## 17 Exceptions

Exception	Condition
System.IO.IOException	An I/O error occurred.

18

19

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 **TextWriter.WriteLine(System.Single) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual void WriteLine(float32 value)  
  
6 [C#]  
7 public virtual void WriteLine(float value)
```

#### 8 **Summary**

9 Writes the text representation of a specified `System.Single` value followed by a line  
10 terminator to the text stream.

#### 11 **Parameters**

Parameter	Description
<i>value</i>	The <code>System.Single</code> value to write.

#### 13 **Description**

14 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
15 `System.IO.TextWriter.WriteLine(value.ToString())`.

#### 16 **Behaviors**

17 As described above.

#### 19 **Exceptions**

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

# 1 TextWriter.WriteLine(System.String, 2 System.Object) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(string format, object  
5 arg0)  
6 [C#]  
7 public virtual void WriteLine(string format, object arg0)
```

## 8 Summary

9 Writes out a formatted string followed by a line terminator to the text stream.

## 10 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The object referenced in the format string.

11

## 12 Description

13 This version of System.IO.TextWriter.WriteLine is equivalent to  
14 System.IO.TextWriter.WriteLine(System.String.Format (*format*, *arg0*)).

15  
16 [*Note:* If the specified object is not referenced in the format string, it is ignored.

17  
18 This method does not search the specified System.String for individual newline  
19 characters (hexadecimal 0x000a) and replace them with  
20 System.IO.TextWriter.NewLine.

21  
22 ]

## 23 Behaviors

24 As described above.

25

## 26 Exceptions

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

<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	<p>The format specification in <i>format</i> is invalid.</p> <p>-or-</p> <p>The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (1).</p>

1

2

# 1 TextWriter.WriteLine(System.String, 2 System.Object, System.Object) Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(string format, object  
5 arg0, object arg1)  
  
6 [C#]  
7 public virtual void WriteLine(string format, object arg0, object arg1)
```

## 8 Summary

9 Writes out a formatted string followed by a line terminator to the text stream.

## 10 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.

11

## 12 Description

13 This version of System.IO.TextWriter.WriteLine is equivalent to  
14 System.IO.TextWriter.WriteLine(System.String.Format (*format*, *arg0*, *arg1*)).

15  
16 [*Note:* If the specified object is not referenced in the format string, it is ignored.

17  
18 This method does not search the specified System.String for individual newline  
19 characters (hexadecimal 0x000a) and replace them with  
20 System.IO.TextWriter.NewLine.

21  
22 ]

## 23 Behaviors

24 As described above.

25

## 26 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	<p>The format specification in <i>format</i> is invalid.</p> <p>-or-</p> <p>The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (2).</p>

1

2

# 1 TextWriter.WriteLine(System.String, 2 System.Object, System.Object, 3 System.Object) Method

```
4 [ILAsm]  
5 .method public hidebysig virtual void WriteLine(string format, object  
6 arg0, object arg1, object arg2)  
  
7 [C#]  
8 public virtual void WriteLine(string format, object arg0, object arg1,  
9 object arg2)
```

## 10 Summary

11 Writes out a formatted string followed by a line terminator to the text stream.

## 12 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.
<i>arg2</i>	The third object referenced in the format string.

## 13 14 Description

15 This version of System.IO.TextWriter.WriteLine is equivalent to  
16 System.IO.TextWriter.WriteLine(System.String.Format (*format*, *arg0*, *arg1*, *arg2*)).

17  
18 If the value of any object parameter is null, it is treated as an empty string.

19  
20 [*Note:* If the specified object is not referenced in the format string, it is ignored.

21  
22 This method does not search the specified System.String for individual newline  
23 characters (hexadecimal 0x000a) and replace them with  
24 System.IO.TextWriter.NewLine.

25  
26 ]

## 27 Behaviors

28 As described above.

1

## 2 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	The format specification in <i>format</i> is invalid.  -or-  The number indicating an argument to be formatted is less than zero, or greater than or equal to the number of provided objects to be formatted (3).

3

4

# 1 `TextWriter.WriteLine(System.String,` 2 `System.Object[])` Method

```
3 [ILAsm]  
4 .method public hidebysig virtual void WriteLine(string format, object[]  
5 arg)  
  
6 [C#]  
7 public virtual void WriteLine(string format, params object[] arg)
```

## 8 Summary

9 Writes out a formatted string followed by a line terminator to the text stream.

## 10 Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> containing the format string.
<i>arg</i>	The array of objects referenced from the format string.

## 11 12 Description

13 This version of `System.IO.TextWriter.WriteLine` is equivalent to  
14 `System.IO.TextWriter.WriteLine(System.String.Format (format, arg))`.

15  
16 [*Note:* If the specified object is not referenced in the format string, it is ignored.

17  
18 This method does not search the specified `System.String` for individual newline  
19 characters (hexadecimal 0x000a) and replace them with  
20 `System.IO.TextWriter.NewLine`.

21  
22 ]

## 23 Behaviors

24 As described above.

25

## 26 Exceptions

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

<b>System.ArgumentNullException</b>	<i>format</i> or <i>arg</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	<p>The format specification in <i>format</i> is invalid.</p> <p>-or-</p> <p>The number indicating an argument to be formatted is less than zero, or greater than or equal to <i>arg</i>.Length.</p>

1

2

1 **The following member must be implemented if the ExtendedNumerics library is present in**  
2 **the implementation.**

### 3 **TextWriter.WriteLine(System.Double) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual void WriteLine(float64 value)  
  
6 [C#]  
7 public virtual void WriteLine(double value)
```

#### 8 **Summary**

9 Writes the text representation of a specified System.Double value followed by a line  
10 terminator to the text stream.

#### 11 **Parameters**

Parameter	Description
<i>value</i>	The System.Double value to write.

#### 13 **Description**

14 This version of System.IO.TextWriter.WriteLine is equivalent to  
15 System.IO.TextWriter.WriteLine (*value*.ToString()).

#### 16 **Behaviors**

17 As described above.

#### 19 **Exceptions**

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.

# 1 TextWriter.Encoding Property

```
2 [ILAsm]  
3 .property class System.Text.Encoding Encoding { public hidebysig virtual  
4 abstract specialname class System.Text.Encoding get_Encoding() }  
5 [C#]  
6 public abstract 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 in which the output is written.

## 11 Behaviors

12 As described above.

13

## 14 Usage

15 This property is required in some XML scenarios where a header must be written  
16 containing the encoding used by the System.IO.TextWriter. This allows XML code to  
17 consume an arbitrary System.IO.TextWriter and generate a correct XML header.

18

19

# 1 TextWriter.FormatProvider Property

```
2 [ILAsm]  
3 .property class System.IFormatProvider FormatProvider { public hidebysig  
4 virtual specialname class System.IFormatProvider get_FormatProvider() }  
  
5 [C#]  
6 public virtual IFormatProvider FormatProvider { get; }
```

## 7 Summary

8 Gets an object that provides formatting for the current instance

## 9 Property Value

10 A System.IFormatProvider object for a specific culture, or the formatting of the current  
11 culture if no other culture is specified.

## 12 Behaviors

13 As described above.

14

15

# 1 TextWriter.NewLine Property

```
2 [ILAsm]
3 .property string NewLine { public hidebysig virtual specialname string
4 get_NewLine() public hidebysig virtual specialname void set_NewLine(string
5 value) }
6 [C#]
7 public virtual string NewLine { get; set; }
```

## 8 Summary

9 Gets or sets the line terminator string used by the current `System.IO.TextWriter`.

## 10 Property Value

11 A `System.String` containing the line terminator string for the current  
12 `System.IO.TextWriter`.

## 13 Description

14 The default line terminator string is `System.Environment.NewLine`.

15  
16 The line terminator string is written to the text stream whenever one of the  
17 `System.IO.TextWriter.WriteLine` methods is called. In order for text written by the  
18 `System.IO.TextWriter` to be readable by a `System.IO.TextReader`, only  
19 `System.Environment.NewLine` should be used as a terminator string. If  
20 `System.IO.TextWriter.NewLine` is set to null, the default newline character is used  
21 instead.

## 22 Behaviors

23 As described above.

24

25