

# System.IO.TextWriter Class

```
[ILAsm]
.class public abstract serializable TextWriter extends
System.MarshalByRefObject implements System.IDisposable

[C#]
public abstract class TextWriter: MarshalByRefObject, IDisposable
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Implements:

- **System.IDisposable**

## Summary

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

## Inherits From: System.MarshalByRefObject

**Library:** BCL

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

## Description

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

[*Note:* By default, a `System.IO.TextWriter` is not thread safe. See `System.IO.TextWriter.Synchronized` for a thread-safe wrapper.]

# TextWriter(System.IFormatProvider)

## Constructor

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

### Summary

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

### Parameters

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

## TextWriter() Constructor

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

### Summary

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

# TextWriter.Null Field

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

## Summary

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

## Description

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

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

## TextWriter.Close() Method

```
[ILAsm]  
.method public hidebysig virtual void Close()  
  
[C#]  
public virtual void Close()
```

### Summary

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

### Description

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

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

]

### Behaviors

As described above.

# TextWriter.Dispose(System.Boolean) Method

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

## Summary

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

## Parameters

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

## Description

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

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

# 1 TextWriter.Flush() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Flush()  
4  
5 [C#]  
6 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.

## 12 Default

13 This method is not implemented in this abstract class.

# TextWriter.Synchronized(System.IO.TextWriter) Method

```
[ILAsm]  
.method public hidebysig static class System.IO.TextWriter  
Synchronized(class System.IO.TextWriter writer)  
  
[C#]  
public static TextWriter Synchronized(TextWriter writer)
```

## Summary

Creates a thread-safe wrapper around the specified `System.IO.TextWriter` instance.

## Parameters

Parameter	Description
<i>writer</i>	The <code>System.IO.TextWriter</code> to synchronize.

## Return Value

A thread-safe `System.IO.TextWriter`.

## Description

This method returns a class that wraps around the specified `System.IO.TextWriter` instance and restricts concurrent access to it by multiple threads.

## Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>writer</i> is null.



# TextWriter.System.IDisposable.Dispose()

## Method

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

## Summary

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

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

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

## Summary

Writes a formatted string to the text stream.

## 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.

## Description

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

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

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

]

## Behaviors

As described above.

## Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<p><i>format</i> is null.</p> <p>-or-</p> <p><i>arg</i> is null.</p>
<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  
3

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

```
[ILAsm]  
.method public hidebysig virtual void Write(string format, object arg0,  
object arg1, object arg2)  
  
[C#]  
public virtual void Write(string format, object arg0, object arg1, object  
arg2)
```

## Summary

Writes out a formatted string to the text stream.

## 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.

## Description

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

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

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

]

## Behaviors

1 As described above.

2

3 **Exceptions**

4

5

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).

6

7

8

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

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

## Summary

Writes out a formatted string to the text stream.

## 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.

## Description

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

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

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

]

## Behaviors

As described above.

## 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>

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

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

## Summary

Writes out the format string to the text stream.

## Parameters

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

## Description

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

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

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

]

## Behaviors

As described above.

## 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  
3

# TextWriter.Write(System.Object) Method

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

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Default

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

## Exceptions

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

- 1
- 2
- 3

# TextWriter.Write(System.String) Method

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

## Summary

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

## Parameters

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

## Description

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

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

## Behaviors

As described above.

## Exceptions

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

- 1
- 2
- 3

**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 <code>System.Decimal</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.

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

## TextWriter.Write(System.Double) Method

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

### Summary

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

### Parameters

Parameter	Description
<i>value</i>	The System.Double 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
System.IO.IOException	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.



# TextWriter.Write(System.UInt64) Method

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

## Summary

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

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

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

## Description

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

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
- 2
- 3

# TextWriter.Write(System.Int64) Method

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

## Summary

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

## Parameters

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

# TextWriter.Write(System.UInt32) Method

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

## Summary

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

## Type Attributes:

- `CLSCompliantAttribute(false)`

## Parameters

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

## Description

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

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
- 2
- 3

# TextWriter.Write(System.Int32) Method

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

## Summary

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

## Parameters

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

# TextWriter.Write(System.Boolean) Method

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

## Summary

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

## Parameters

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

# TextWriter.Write(System.Char[], System.Int32, System.Int32) Method

```
[ILAsm]
.method public hidebysig virtual void Write(char[] buffer, int32 index,
int32 count)

[C#]
public virtual void Write(char[] buffer, int index, int count)
```

## Summary

Writes a subarray of characters to the text stream.

## 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.

## Description

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

## Behaviors

As described above.

## Default

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



## 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.

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

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

## Summary

Writes a character array to the text stream.

## 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.

## Description

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

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

## Behaviors

As described above.

## Exceptions

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

- 1
- 2
- 3

# TextWriter.Write(System.Char) Method

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

## Summary

Writes a character to the text stream.

## Parameters

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

## Behaviors

As described above.

## Default

This method is not implemented in this abstract class.

## 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.WriteLine(System.Decimal) Method

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

### Summary

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

### Parameters

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

### Description

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

### Behaviors

As described above.

### Exceptions

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



# TextWriter.WriteLine(System.Int64) Method

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

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Exceptions

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

# TextWriter.WriteLine(System.UInt32)

## Method

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

### Summary

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

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

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

### Description

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

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

### Behaviors

As described above.

### Exceptions



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

1  
2  
3

# TextWriter.WriteLine(System.Int32) Method

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

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Exceptions

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

# TextWriter.WriteLine(System.Boolean)

## Method

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

### Summary

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

### Parameters

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

### Description

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

### Behaviors

As described above.

### Exceptions

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

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

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

## Summary

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

## Parameters

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

## Description

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

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

## Behaviors

As described above.

## 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.

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

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

## Summary

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

## Parameters

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

## Description

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

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

## Behaviors

As described above.

## Exceptions

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

- 1
- 2
- 3

# TextWriter.WriteLine(System.Char) Method

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

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Exceptions

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



# TextWriter.WriteLine() Method

```
[ILAsm]  
.method public hidebysig virtual void WriteLine()  
  
[C#]  
public virtual void WriteLine()
```

## Summary

Writes a line terminator to the text stream.

## Description

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

## Behaviors

As described above.

## Exceptions

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

# TextWriter.WriteLine(System.UInt64)

## Method

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

### Summary

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

### Type Attributes:

- `CLSCompliantAttribute(false)`

### Parameters

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

### Description

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

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

### Behaviors

As described above.

### Exceptions

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

1  
2  
3

# TextWriter.WriteLine(System.String) Method

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

## Summary

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

## 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.

## Description

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

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

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

## Behaviors

As described above.

## Exceptions

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

**System.IO.IOException**

An I/O error occurred.

1

2

3

# TextWriter.WriteLine(System.Object) Method

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

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Exceptions

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

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

## TextWriter.WriteLine(System.Single) Method

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

### Summary

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

### Parameters

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

### Description

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

### Behaviors

As described above.

### Exceptions

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

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

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

## Summary

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

## Parameters

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

## Description

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

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

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

]

## Behaviors

As described above.

## 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  
3

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

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

## Summary

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

## 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.

## Description

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

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

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

]

## Behaviors

As described above.

## 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>

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

```
[ILAsm]
.method public hidebysig virtual void WriteLine(string format, object
arg0, object arg1, object arg2)

[C#]
public virtual void WriteLine(string format, object arg0, object arg1,
object arg2)
```

## Summary

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

## 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.

## Description

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

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

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

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

]

## Behaviors

1 As described above.

2

3 **Exceptions**

4

5

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).

6

7

8

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

```
[ILAsm]
.method public hidebysig virtual void WriteLine(string format, object[]
arg)

[C#]
public virtual void WriteLine(string format, params object[] arg)
```

## Summary

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

## 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.

## Description

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

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

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

]

## Behaviors

As described above.

## 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  
3

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

## TextWriter.WriteLine(System.Double) Method

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

### Summary

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

### Parameters

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

### Description

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

### Behaviors

As described above.

### Exceptions

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



# TextWriter.Encoding Property

```
[ILAsm]  
.property class System.Text.Encoding Encoding { public hidebysig virtual  
abstract specialname class System.Text.Encoding get_Encoding() }  
  
[C#]  
public abstract Encoding Encoding { get; }
```

## Summary

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

## Property Value

The `System.Text.Encoding` in which the output is written.

## Behaviors

As described above.

## Usage

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

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

## 13 Behaviors

14 As described above.  
15  
16

## TextWriter.NewLine Property

```
[ILAsm]
.property string NewLine { public hidebysig virtual specialname string
get_NewLine() public hidebysig virtual specialname void set_NewLine(string
value) }

[C#]
public virtual string NewLine { get; set; }
```

### Summary

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

### Property Value

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

### Description

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

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

### Behaviors

As described above.