

# 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]
- *Version:* 1.0.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.]

# 1 TextWriter() Constructor

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

## 6 Summary

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

9

# 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  
10 **System.IO.TextWriter** class with the specified format provider.

### 11 Parameters

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

14  
15  
16

# 1 TextWriter.Null Field

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

## 7 Summary

8 Provides a **System.IO.TextWriter** object without a backing store.

## 9 Description

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

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

16

# 1 TextWriter.Close() Method

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

## 6 Summary

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

## 9 Description

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

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

## 15 Behaviors

16 As described above.

17

# 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>	<b>true</b> to release both managed and unmanaged resources; <b>false</b> 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** may 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  
8 for the current **System.IO.TextWriter** instance.

## 9 Behaviors

10 As described above.

## 11 Default

12 This method is not implemented in this abstract class.

13

# 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 <b>System.IO.TextWriter</b> 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
<b>System.ArgumentNullException</b>	<i>writer</i> is <b>null</b> .



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

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

## 8 Summary

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

11

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

```
[ILASM]
.method public hidebysig virtual void Write(string format,
class System.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 <b>System.String</b> 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>	<i>format</i> is <b>null</b> .
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.FormatException</b>	The format specification in <i>format</i> is invalid.

1  
2  
3

	<p>-or-</p> <p>The number indicating an argument to be formatted is less than zero, or greater than or equal to <i>arg.Length</i>.</p>
--	--

# TextWriter.Write(System.String, 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 <b>System.String</b> 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

As described above.

## Exceptions

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

<b>System.ArgumentNullException</b>	<p><i>format</i> is <b>null</b>.</p> <p>-or-</p> <p>An object referenced in the <i>format</i> string is <b>null</b>.</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 the number of provided objects to be formatted (3).</p>

1  
2  
3

# 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 <b>System.String</b> 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 <b>null</b> .

1  
2  
3

	<p>-or-</p> <p>An object referenced in the <i>format</i> string is <b>null</b>.</p>
<b>System.IO.IOException</b>	<p>An I/O error occurred.</p>
<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 **TextWriter.Write(System.String,**

## 2 **System.Object) Method**

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

### 8 **Summary**

9 Writes out the format string to the text stream.

### 10 **Parameters**

11  
12

Parameter	Description
<i>format</i>	A <b>System.String</b> containing the format string.
<i>arg0</i>	An 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*,  
17 *arg0*)).

18

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

21

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

### 25 **Behaviors**

26 As described above.

### 27 **Exceptions**

28

29

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



	An object referenced in the <i>format</i> string is <b>null</b> .
<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 <b>null</b> , 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.

# 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 <b>System.String</b> to write. If <i>value</i> is <b>null</b> , nothing is written.

## Description

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

[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
<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.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 <b>System.Decimal</b> 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 <b>System.Double</b> 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.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 <b>System.Single</b> 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 <b>System.UInt64</b> 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.

# 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 <b>System.Int64</b> 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 <b>System.UInt32</b> 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.

# 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 <b>System.Int32</b> 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.Boolean)**

## 2 **Method**

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

### 7 **Summary**

8 Writes the text representation of a specified **System.Boolean** value to  
9 the text stream.

### 10 **Parameters**

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

### 14 **Description**

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

### 17 **Behaviors**

18 As described above.

### 19 **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(class
System.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 <b>null</b> .
<b>System.ArgumentException</b>	$(index + count) > buffer.Length$ .
<b>System.ArgumentOutOfRangeException</b>	$index < 0$ or $count < 0$ .

1  
2  
3

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

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

```
2 [ILASM]
3 .method public hidebysig virtual void Write(class
4 System.Char[] buffer)

5 [C#]
6 public virtual void Write(char[] buffer)
```

## 7 Summary

8 Writes a character array to the text stream.

## 9 Parameters

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

## 13 Description

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

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

## 20 Behaviors

21 As described above.

## 22 Exceptions

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

# 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 <b>System.Char</b> 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.

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

```
[ILASM]
.method public hidebysig virtual void WriteLine(string
format, class System.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 <b>System.String</b> 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 <b>null</b> .
<b>System.IO.IOException</b>	An I/O error occurred.



1  
2  
3

<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.Length</i>.</p>
-------------------------------	---

# TextWriter.WriteLine(System.String, 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 <b>System.String</b> 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))**.

[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>	<p><i>format</i> is <b>null</b>.</p> <p>-or-</p> <p>An object referenced in the <i>format</i> string is <b>null</b>.</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 the number of provided objects to be formatted (3).</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 <b>System.String</b> 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 <b>null</b> .

1  
2  
3

	<p>-or-</p> <p>An object referenced in the <i>format</i> string is <b>null</b>.</p>
<b>System.IO.IOException</b>	<p>An I/O error occurred.</p>
<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) 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 <b>System.String</b> 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 <b>null</b> . -or-

	An object referenced in the <i>format</i> string is <b>null</b> .
<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.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 <b>null</b> , 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
<b>System.IO.IOException</b>	An I/O error occurred.



# 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 <b>System.String</b> to write. If <i>value</i> is <b>null</b> , only the line terminator is written.

### Description

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

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
<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 <b>System.Decimal</b> 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.

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 <b>System.Double</b> 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.

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 <b>System.Single</b> 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.

# 1 **TextWriter.WriteLine(System.UInt64)**

## 2 **Method**

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

### 8 **Summary**

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

### 11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

### 13 **Parameters**

14  
15

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

16

### 17 **Description**

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

20

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

### 23 **Behaviors**

24 As described above.

### 25 **Exceptions**

26

27

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

28

29

30

# 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 <b>System.Int64</b> 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 <b>System.UInt32</b> 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.

# 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 <b>System.Int32</b> 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.



# 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  
9 a line terminator to the text stream.

### 10 Parameters

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

### 13 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.

### 19 Exceptions

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

22  
23  
24

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

```
[ILASM]
.method public hidebysig virtual void WriteLine(class
System.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 <b>System.Char</b> array from which characters are read.
<i>index</i>	A <b>System.Int32</b> that specifies the index in <i>buffer</i> at which to begin reading.
<i>count</i>	A <b>System.Int32</b> 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 <b>null</b> .
<b>System.ArgumentException</b>	$(index + count) > buffer$ .

1  
2  
3

<b>System.ArgumentOutOfRangeException</b>	<i>index</i> < 0 -or- <i>count</i> < 0.
<b>System.IO.IOException</b>	An I/O error occurred.

# TextWriter.WriteLine(System.Char[])

## Method

```
[ILASM]
.method public hidebysig virtual void WriteLine(class
System.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 <b>System.Char</b> array from which data is read. If <i>buffer</i> is <b>null</b> , 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
<b>System.IO.IOException</b>	An I/O error occurred.

# 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 <b>System.Char</b> 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.

# 1 TextWriter.WriteLine() Method

```
2 [ILASM]
3 .method public hidebysig virtual void WriteLine()
4
5 [C#]
6 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.To**  
11 **CharArray())**.

## 12 Behaviors

13 As described above.

## 14 Exceptions

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

# 1 TextWriter.Encoding Property

```
2 [ILASM]
3 .property class System.Text.Encoding Encoding { public
4 hidebysig virtual abstract specialname class
5 System.Text.Encoding get_Encoding() }

6 [C#]
7 public abstract Encoding Encoding { get; }
```

## 8 Summary

9 Gets the **System.Text.Encoding** in which the output is written.

## 10 Property Value

11

12 The **System.Text.Encoding** in which the output is written.

## 13 Behaviors

14 As described above.

## 15 Usage

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

20

# 1 TextWriter.FormatProvider Property

```
2 [ILASM]
3 .property class System.IFormatProvider FormatProvider {
4 public hidebysig virtual specialname class
5 System.IFormatProvider get_FormatProvider() }

6 [C#]
7 public virtual IFormatProvider FormatProvider { get; }
```

## 8 Summary

9 Gets an object that provides formatting for the current instance

## 10 Property Value

11

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

## 14 Behaviors

15 As described above.

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 "\n" or **System.Environment.NewLine** should be used as terminator strings. If **System.IO.TextWriter.NewLine** is set to **null**, the default newline character is used instead.

## Behaviors

As described above.