

# System.Console Class

```
[ILAsm]
.class public sealed Console extends System.Object

[C#]
public sealed class Console
```

## 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)

## Summary

Represents the standard input, output, and error streams for console applications.

## Inherits From: System.Object

**Library:** BCL

**Thread Safety:** This type is safe for multithreaded operations.

## Description

The `System.Console` class provides basic input and output support for applications that read from and write characters to the console. If the console does not exist, as in a GUI application, writing to the console produces no result, and no exception is raised.

The standard input, output, and error streams are represented by properties, and are automatically associated with the console when the application starts. Applications can redirect these properties to other streams; for example, streams associated with files instead of the console. [*Note:* For additional information see the `System.Console.SetIn`, `System.Console.SetOut`, and `System.Console.SetError` methods.]

By default, the read methods in this class use the standard input stream and the write methods use the standard output stream.

The write methods support writing data with or without automatically appending carriage return and linefeed characters. This enables the writing of strings, formatted strings, arrays of characters, instances of primitive types, and arbitrary objects without first having to convert them to strings.

This class uses synchronized `System.IO.TextReader` and `System.IO.TextWriter` instances. Multiple threads can concurrently read from and/or write to an instance of this type.

## 1   **Example**

2       The following example demonstrates the use of basic `System.Console` input and output  
3       functions. The program waits for the user to enter a name.

4  
5       [C#]

```
6  
7   using System;  
8  
9   public class ConsoleTest {  
10       public static void Main() {  
11           Console.Write("Hello ");  
12           Console.WriteLine("World!");  
13           Console.Write("What is your name: ");  
14           string name = Console.ReadLine();  
15           Console.Write("Hello, ");  
16           Console.Write(name);  
17           Console.WriteLine("!");  
18       }  
19   }
```

20   The output for a user who entered the name "Fred" is

21  
22   Hello World!

23  
24  
25   What is your name: Fred

26  
27  
28   Hello, Fred!

29

30

# 1 Console.OpenStandardError() Method

```
2 [ILAsm]  
3 .method public hidebysig static class System.IO.Stream OpenStandardError()  
4 [C#]  
5 public static Stream OpenStandardError()
```

## 6 Summary

7 Returns the standard error stream.

## 8 Return Value

9 A new synchronized `System.IO.Stream` object that writes to the console.

10

# Console.OpenStandardError(System.Int32)

## Method

```
[ILAsm]  
.method public hidebysig static class System.IO.Stream  
OpenStandardError(int32 bufferSize)  
  
[C#]  
public static Stream OpenStandardError(int bufferSize)
```

### Summary

Returns the standard error stream.

### Parameters

Parameter	Description
<i>bufferSize</i>	A <code>System.Int32</code> that specifies the desired internal stream buffer size.

### Return Value

A new synchronized `System.IO.Stream` object that writes to the console.

### Description

Buffering console streams is not required to be supported. If it is not supported, the *bufferSize* parameter is ignored, and this method behaves identically to `System.Console.OpenStandardError()`. If buffering is supported, the buffering behavior of the `System.Console` class is implementation-specific.

# 1 Console.OpenStandardInput() Method

```
2 [ILAsm]  
3 .method public hidebysig static class System.IO.Stream OpenStandardInput()  
4 [C#]  
5 public static Stream OpenStandardInput()
```

## 6 Summary

7 Returns the standard input stream.

## 8 Return Value

9 A new synchronized `System.IO.Stream` object that reads from the console.

10

# Console.OpenStandardInput(System.Int32)

## Method

```
[ILAsm]  
.method public hidebysig static class System.IO.Stream  
OpenStandardInput(int32 bufferSize)  
  
[C#]  
public static Stream OpenStandardInput(int bufferSize)
```

### Summary

Returns the standard input stream.

### Parameters

Parameter	Description
<i>bufferSize</i>	A <code>System.Int32</code> that specifies the desired internal stream buffer size.

### Return Value

A new synchronized `System.IO.Stream` object that reads from the console.

### Description

Buffering console streams is not required to be supported. If it is not supported, the *bufferSize* parameter is ignored, and this method behaves identically to `System.Console.OpenStandardInput()`. If buffering is supported, the buffering behavior of the `System.Console` class is implementation-specific.

# 1 Console.OpenStandardOutput() Method

```
2 [ILAsm]  
3 .method public hidebysig static class System.IO.Stream  
4 OpenStandardOutput()  
  
5 [C#]  
6 public static Stream OpenStandardOutput()
```

## 7 Summary

8 Returns the standard output stream.

## 9 Return Value

10 A new synchronized `System.IO.Stream` object that writes to the console.

# Console.OpenStandardOutput(System.Int32)

## Method

```
[ILAsm]  
.method public hidebysig static class System.IO.Stream  
OpenStandardOutput(int32 bufferSize)  
  
[C#]  
public static Stream OpenStandardOutput(int bufferSize)
```

### Summary

Returns the standard output stream. The desired size of the internal buffer for the stream is specified.

### Parameters

Parameter	Description
<i>bufferSize</i>	A <code>System.Int32</code> that specifies the desired internal stream buffer size.

### Return Value

A new synchronized `System.IO.Stream` object that writes to the console.

### Description

Buffering console streams is not required to be supported. If it is not supported, the *bufferSize* parameter is ignored, and this method behaves identically to `System.Console.OpenStandardOutput()`. If buffering is supported, the buffering behavior of the `System.Console` class is implementation-specific.



# Console.Read() Method

```
[ILAsm]  
.method public hidebysig static int32 Read()  
  
[C#]  
public static int Read()
```

## Summary

Reads the next character from the standard input stream.

## Return Value

The next character from the character source represented as a *System.Int32*, or -1 if at the end of the stream.

## Description

This method will not return until the read operation is terminated; for example, by the user pressing the enter key. If data is available, the input stream contains what the user entered, suffixed with the environment dependent newline character.

## Exceptions

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

# Console.ReadLine() Method

```
[ILAsm]  
.method public hidebysig static string ReadLine()  
  
[C#]  
public static string ReadLine()
```

## Summary

Reads the next line of characters from the `System.Console.In` `System.IO.TextReader`.

## Return Value

A `System.String` containing the next line from the input stream, or `null` if the end of the input stream has already been reached.

## Description

A line is defined as a sequence of characters followed by a carriage return (Unicode 0x000d), a line feed (Unicode 0x000a), or a `System.Environment.NewLine`. The returned string does not contain the terminating character(s).

## Exceptions

Exception	Condition
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.OutOfMemoryException</b>	There is insufficient memory to allocate a buffer for the returned string.

# Console.SetError(System.IO.TextWriter) Method

```
[ILAsm]  
.method public hidebysig static void SetError(class System.IO.TextWriter  
newError)  
  
[C#]  
public static void SetError(TextWriter newError)
```

## Summary

Sets the `System.Console.Error` property to the specified `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>newError</i>	A <code>System.IO.TextWriter</code> that becomes the new standard error output stream.

## Description

This method replaces the `System.Console.Error` property with a synchronized `System.IO.TextWriter` returned by `System.IO.TextWriter.Synchronized( newError )`.

[*Note:* By default, the `System.Console.Error` property is set to the system's standard error stream.]

## Exceptions

Exception	Condition
<b>System.Security.SecurityException</b>	The caller does not have the required permission.
<b>System.ArgumentNullException</b>	<i>newError</i> is null.

## Permissions

Permission	Description
<b>System.Security.Permissions.</b>	Requires

<b>SecurityPermissionFlag</b>	System.Security.Permissions.SecurityPermissionFlag. UnmanagedCode permission.
-------------------------------	--

1

2

# Console.SetIn(System.IO.TextReader)

## Method

```
[ILAsm]  
.method public hidebysig static void SetIn(class System.IO.TextReader  
newIn)  
  
[C#]  
public static void SetIn(TextReader newIn)
```

### Summary

Sets the `System.Console.In` property to the specified `System.IO.TextReader`.

### Parameters

Parameter	Description
<i>newIn</i>	A <code>System.IO.TextReader</code> that becomes the new standard input stream.

### Description

This method replaces the `System.Console.In` property with a synchronized `System.IO.TextReader` returned by `System.IO.TextReader.Synchronized( newIn )`.

[*Note:* By default, the `System.Console.In` property is set to the system's standard input stream.]

### Exceptions

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

### Permissions

Permission	Description
<b>System.Security.Permissions.SecurityPermissionFlag</b>	Requires permission to execute unmanaged code. See <code>System.Security.Permissions.SecurityPermissionFlag.UnmanagedCode</code> .



# Console.SetOut(System.IO.TextWriter)

## Method

```
[ILAsm]  
.method public hidebysig static void SetOut(class System.IO.TextWriter  
newOut)  
  
[C#]  
public static void SetOut(TextWriter newOut)
```

### Summary

Sets the `System.Console.Out` property to the specified `System.IO.TextWriter`.

### Parameters

Parameter	Description
<i>newOut</i>	A <code>System.IO.TextWriter</code> that becomes the new standard output stream.

### Description

This method replaces the `System.Console.Out` property with a synchronized `System.IO.TextWriter` returned by `System.IO.TextWriter.Synchronized(newOut)`.

[*Note:* By default, the `System.Console.Out` property is set to the system's standard output stream.]

### Exceptions

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

### Permissions

Permission	Description
<b>System.Security.Permissions.SecurityPermissionFlag</b>	Requires permission to execute unmanaged code. See <code>System.Security.Permissions.SecurityPermissionFlag.UnmanagedCode</code> .





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

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

## Summary

Writes a formatted string to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies the format string.
<i>arg</i>	An array of objects referenced in the format string.

## Description

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

For more information on format strings, see the `System.String` class overview.

]

## 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>	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> .



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

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

## Summary

Writes a formatted string to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies 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

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

[*Note:* For more information on format strings see the `System.String` class overview.]

## 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 (3).</p>
-------------------------------	--

1

2

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

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

## Summary

Writes a formatted string to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies 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

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

For more information on format strings, see the `System.String` class overview.

]

## 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 (2).
--	---

1

2

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

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

## Summary

Writes a formatted string to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

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

For more information on format strings, see the `System.String` class overview.

]

## 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 (1).

1

2



# Console.Write(System.Boolean) Method

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

## Summary

Writes the text representation of a System.Boolean to the System.Console.OutSystem.IO.TextWriter.

## Parameters

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

## Description

This version of System.Console.Write is equivalent to System.Console.Out.Write(*value*.ToString () ).

## Exceptions

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

# Console.Write(System.Char) Method

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

## Summary

Writes a character to the System.Console.OutSystem.IO.TextWriter.

## Parameters

Parameter	Description
<i>value</i>	The Unicode character to write to the text stream.

## Exceptions

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

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

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

## Summary

Writes a character array to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(buffer)`.

## Exceptions

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

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

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

## Summary

Writes a subarray of characters to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(buffer, index, count)`.

## Exceptions

Exception	Condition
<b>System.ArgumentException</b>	( <i>index</i> + <i>count</i> ) is greater than the length of <i>buffer</i> .
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> or <i>count</i> is negative.
<b>System.ArgumentNullException</b>	<i>buffer</i> is null.

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

## Console.Write(System.Double) Method

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

### Summary

Writes the text representation of a specified System.Double to the System.Console.OutSystem.IO.TextWriter.

### Parameters

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

### Description

This version of System.Console.Write is equivalent to System.Console.Out.Write (*value.ToString()*).

### 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.**

## Console.Write(System.Decimal) Method

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

### Summary

Writes the text representation of a specified System.Decimal to the System.Console.OutSystem.IO.TextWriter.

### Parameters

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

### Description

This version of System.Console.Write is equivalent to System.Console.Out.Write (*value.ToString()*).

### 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.**

## Console.Write(System.Single) Method

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

### Summary

Writes the text representation of a specified `System.Single` to the `System.Console.Out` `System.IO.TextWriter`.

### Parameters

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

### Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write` (*value.ToString()*).

### Exceptions

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

# Console.Write(System.Int32) Method

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

## Summary

Writes the text representation of a specified `System.Int32` to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(value.ToString())`.

## Exceptions

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



# Console.Write(System.UInt32) Method

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

## Summary

Writes the text representation of a specified System.UInt32 to the System.Console.OutSystem.IO.TextWriter.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

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

## Description

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

This version of System.Console.Write is equivalent to System.Console.Out.Write(*value*.ToString()).

## Exceptions

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

# Console.Write(System.Int64) Method

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

## Summary

Writes the text representation of a specified `System.Int64` to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(value.ToString())`.

## Exceptions

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

# Console.Write(System.UInt64) Method

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

## Summary

Writes the text representation of a specified System.UInt64 to the System.Console.OutSystem.IO.TextWriter.

## Type Attributes:

- CLSCompliantAttribute(false)

## Parameters

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

## Description

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

This version of System.Console.Write is equivalent to System.Console.Out.Write(*value*.ToString()).

## Exceptions

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

# Console.Write(System.Object) Method

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

## Summary

Writes the text representation of a specified object to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>value</i>	The object to write. If <i>value</i> is null, <code>System.String.Empty</code> is written.

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(value.ToString())`.

[Note: If *value* is null, no exception is thrown and nothing is written. Otherwise, the object's `System.Object.ToString` method is called to produce the string representation, and the resulting string is written to the output stream.]

## Exceptions

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

# Console.Write(System.String) Method

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

## Summary

Writes a specified string to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.Write` is equivalent to `System.Console.Out.Write(value)`.

[*Note:* If specified value is null, nothing is written to the output stream.]

## Exceptions

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

# Console.WriteLine(System.Boolean) Method

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

## Summary

Writes the text representation of a `System.Boolean` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString() )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

## Console.WriteLine() Method

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

### Summary

Writes a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

### Description

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

### Exceptions

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

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

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

## Summary

Writes a formatted string and a new line to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies the format string.
<i>arg</i>	An array of objects referenced in the format string.

## Description

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

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

For more information on format strings, see the `System.String` class overview.

]

## 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>	The format specification in <i>format</i> is invalid.



	<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>
--	--

1

2

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

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

## Summary

Writes a formatted string and a new line to the  
System.Console.OutSystem.IO.TextWriter.

## Parameters

Parameter	Description
<i>format</i>	A System.String that specifies 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

The default line terminator is the value of the System.Environment.NewLine property. The line terminator can be set using the System.IO.TextWriter.NewLine property of the System.Console.Out stream.

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

For more information on format strings, see the System.String class overview.

]

## 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 (3).</p>

1

2

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

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

## Summary

Writes a formatted string and a new line to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>format</i>	A <code>System.String</code> that specifies 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

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

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

For more information on format strings, see the `System.String` class overview.

]

## 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

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

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

## Summary

Writes a formatted string and a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

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

For more information on format strings, see the `System.String` class overview.

]

## 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 (1).
--	--

1

2

# Console.WriteLine(System.String) Method

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

## Summary

Writes a specified `System.String` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## 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.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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



# Console.WriteLine(System.Object) Method

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

## Summary

Writes the text representation of a specified object followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## 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.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString() )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.WriteLine(System.UInt64) Method

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

## Summary

Writes the text representation of a specified `System.UInt64` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## 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.Console.WriteLine(System.Decimal)`.

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString())`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.WriteLine(System.Int64) Method

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

## Summary

Writes the text representation of a specified `System.Int64` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString() )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.WriteLine(System.UInt32) Method

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

## Summary

Writes the text representation of a specified `System.UInt32` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## 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.Console.WriteLine(System.Int64)`.

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString() )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.WriteLine(System.Int32) Method

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

## Summary

Writes the text representation of a specified `System.Int32` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

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

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value.ToString ()`).

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## 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.**

## Console.WriteLine(System.Single) Method

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

### Summary

Writes the text representation of a specified `System.Single` followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

### Parameters

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

### Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine (value.ToString ())`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

### 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.**

## Console.WriteLine(System.Double) Method

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

### Summary

Writes the text representation of a specified System.Double followed by a line terminator to the System.Console.OutSystem.IO.TextWriter.

### Parameters

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

### Description

This version of System.Console.WriteLine is equivalent to System.Console.Out.WriteLine (*value*.ToString ()).

The default line terminator is the value of theSystem.Environment.NewLine property. The line terminator can be set using theSystem.IO.TextWriter.NewLine property of the System.Console.Out stream.

### 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.**

## Console.WriteLine(System.Decimal) Method

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

### Summary

Writes the text representation of a specified System.Decimal followed by a line terminator to the System.Console.OutSystem.IO.TextWriter.

### Parameters

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

### Description

This version of System.Console.WriteLine is equivalent to System.Console.Out.WriteLine (*value*.ToString ()).

The default line terminator is the value of the System.Environment.NewLine property. The line terminator can be set using the System.IO.TextWriter.NewLine property of the System.Console.Out stream.

### Exceptions

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



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

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

## Summary

Writes a subarray of characters followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>buffer</i>	The Unicode character array from which data is read.
<i>index</i>	A <code>System.Int32</code> that specifies the index into <i>buffer</i> at which to begin reading.
<i>count</i>	A <code>System.Int32</code> that specifies the number of characters to write.

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( buffer, index, count )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

Exception	Condition
<b>System.ArgumentException</b>	( <i>index</i> + <i>count</i> ) is greater than the length of <i>buffer</i> .
<b>System.IO.IOException</b>	An I/O error occurred.
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> or <i>count</i> is negative.

<b>System.ArgumentNullException</b>	<i>buffer</i> is null.
-------------------------------------	------------------------

1

2

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

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

## Summary

Writes an array of characters followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>buffer</i>	The Unicode character array to write. If <i>buffer</i> is null, only the line terminator is written.

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( buffer )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.WriteLine(System.Char) Method

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

## Summary

Writes a character followed by a line terminator to the `System.Console.Out` `System.IO.TextWriter`.

## Parameters

Parameter	Description
<i>value</i>	The Unicode character to write to the text stream.

## Description

This version of `System.Console.WriteLine` is equivalent to `System.Console.Out.WriteLine( value )`.

The default line terminator is the value of the `System.Environment.NewLine` property. The line terminator can be set using the `System.IO.TextWriter.NewLine` property of the `System.Console.Out` stream.

## Exceptions

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

# Console.Error Property

```
[ILAsm]  
.property class System.IO.TextWriter Error { public hidebysig static  
specialname class System.IO.TextWriter get_Error() }  
  
[C#]  
public static TextWriter Error { get; }
```

## Summary

Gets the system's standard error output stream.

## Property Value

A synchronized `System.IO.TextWriter` object where error output is sent.

## Description

This property is read-only.

This property can be redirected using the `System.Console.SetError` method.

[*Note:* If the application does not have a `System.Console`, `System.Console.Error` behaves like `System.IO.TextWriter.Null`.]

# Console.In Property

```
[ILAsm]  
.property class System.IO.TextReader In { public hidebysig static  
specialname class System.IO.TextReader get_In() }  
  
[C#]  
public static TextReader In { get; }
```

## Summary

Gets the system's standard input stream.

## Property Value

A synchronized `System.IO.TextReader` object from which user input is received.

## Description

This property is read-only.

This property can be redirected using the `System.Console.SetIn` method.

[*Note:* If the application does not have a `System.Console`, `System.Console.In` behaves like `System.IO.TextReader.Null`.]

# Console.Out Property

```
[ILAsm]  
.property class System.IO.TextWriter Out { public hidebysig static  
specialname class System.IO.TextWriter get_Out() }  
  
[C#]  
public static TextWriter Out { get; }
```

## Summary

Gets the system's standard output stream.

## Property Value

A synchronized `System.IO.TextWriter` object where normal output is sent.

## Description

This property is read-only.

This property can be redirected using the `System.Console.SetOut` method.

[*Note:* If the application does not have a `System.Console`, `System.Console.Out` behaves like `System.IO.TextWriter.Null`.]