

# 1 System.Console Class

```
2 [ILAsm]  
3 .class public sealed Console extends System.Object  
4 [C#]  
5 public sealed class Console
```

## 6 Assembly Info:

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

## 12 Summary

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

## 14 Inherits From: System.Object

15

16 **Library:** BCL

17

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

19

## 20 Description

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

24

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

31

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

34

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

39

40 This class uses synchronized `System.IO.TextReader` and `System.IO.TextWriter`  
41 instances. Multiple threads can concurrently read from and/or write to an instance of  
42 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

# 1 Console.OpenStandardError(System.Int32)

## 2 Method

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

### 8 Summary

9 Returns the standard error stream.

### 10 Parameters

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

11

### 12 Return Value

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

### 14 Description

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

19

# 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

# 1 Console.OpenStandardInput(System.Int32)

## 2 Method

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

### 8 Summary

9 Returns the standard input stream.

### 10 Parameters

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

11

### 12 Return Value

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

### 14 Description

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

19

# 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.

11

# 1 Console.OpenStandardOutput(System.Int32)

## 2 Method

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

### 8 Summary

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

### 11 Parameters

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

### 12

### 13 Return Value

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

### 15 Description

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

20

# 1 Console.Read() Method

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

## 6 Summary

7 Reads the next character from the standard input stream.

## 8 Return Value

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

## 11 Description

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

## 15 Exceptions

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

16

17

# 1 Console.ReadLine() Method

```
2 [ILAsm]  
3 .method public hidebysig static string ReadLine()  
4 [C#]  
5 public static string ReadLine()
```

## 6 Summary

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

## 8 Return Value

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

## 11 Description

12 A line is defined as a sequence of characters followed by a carriage return (Unicode  
13 0x000d), a line feed (Unicode 0x000a), or a `System.Environment.NewLine`. The  
14 returned string does not contain the terminating character(s).

## 15 Exceptions

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

16

17

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

## 2 Method

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

### 8 Summary

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

### 10 Parameters

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

### 11 Description

13 This method replaces the `System.Console.Error` property with a synchronized  
14 `System.IO.TextWriter` returned by `System.IO.TextWriter.Synchronized( newError`  
15 `)`.

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

### 21 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.

### 22 Permissions

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

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

1

2

# 1 Console.SetIn(System.IO.TextReader)

## 2 Method

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

### 8 Summary

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

### 10 Parameters

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

### 11 Description

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

15  
16 [*Note:* By default, the `System.Console.In` property is set to the system's standard  
17 input stream.]

### 20 Exceptions

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

### 21 Permissions

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



# 1 Console.SetOut(System.IO.TextWriter)

## 2 Method

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

### 8 Summary

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

### 10 Parameters

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

### 11 Description

13 This method replaces the `System.Console.Out` property with a synchronized  
14 `System.IO.TextWriter` returned by `System.IO.TextWriter.Synchronized( newOut )`.  
15  
16 [Note: By default, the `System.Console.Out` property is set to the system's standard  
17 output stream.]  
18  
19

### 20 Exceptions

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

### 21 Permissions

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



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

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

## 7 Summary

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

## 9 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.

10

## 11 Description

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

13

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

15

16 ]

## 17 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> .

18



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

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

## 9 Summary

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

## 11 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.

## 12 Description

14 [Note: If a specified object is not referenced in the format string, it is ignored.]  
15  
16  
17

18 [Note: For more information on format strings see the `System.String` class overview.]  
19  
20

## 21 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.

**System.FormatException**

The format specification in *format* 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).

1

2

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

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

## 8 Summary

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

## 10 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.

11

## 12 Description

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

14

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

16

17 ]

## 18 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

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

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

## 7 Summary

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

## 9 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.

10

## 11 Description

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

13

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

15

16 ]

## 17 Exceptions

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

1

2

# 1 Console.Write(System.Boolean) Method

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

## 6 Summary

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

## 9 Parameters

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

10

## 11 Description

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

## 14 Exceptions

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

15

16

# 1 Console.Write(System.Char) Method

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

## 6 Summary

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

## 8 Parameters

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

9

## 10 Exceptions

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

11

12

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

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

## 6 Summary

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

## 8 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.

9

## 10 Description

11 This version of System.Console.Write is equivalent to System.Console.Out.Write(  
12 *buffer* ).

## 13 Exceptions

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

14

15

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

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

## 8 Summary

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

## 10 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.

## 11 12 Description

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

## 15 Exceptions

Exception	Condition
<b>System.ArgumentException</b>	<i>(index + 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.

16  
17

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

### 3 Console.Write(System.Double) Method

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

#### 8 Summary

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

#### 11 Parameters

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

#### 12 13 Description

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

#### 16 Exceptions

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

17  
18

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

### 3 Console.Write(System.Decimal) Method

```
4 [ILAsm]  
5 .method public hidebysig static void Write(decimal value)  
6  
7 [C#]  
8 public static void Write(decimal value)
```

#### 8 Summary

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

#### 11 Parameters

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

#### 12 13 Description

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

#### 16 Exceptions

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

17  
18

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

### 3 Console.Write(System.Single) Method

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

#### 8 Summary

9 Writes the text representation of a specified System.Single to the  
10 System.Console.OutSystem.IO.TextWriter.

#### 11 Parameters

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

#### 12 13 Description

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

#### 16 Exceptions

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

17  
18

# 1 Console.Write(System.Int32) Method

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

## 6 Summary

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

## 9 Parameters

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

10

## 11 Description

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

## 14 Exceptions

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

15

16

# 1 Console.WriteLine(System.UInt32) Method

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

## 6 Summary

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

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

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

12

## 13 Description

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

16  
17 This version of System.Console.WriteLine is equivalent to System.Console.Out.WriteLine(  
18 *value*.ToString()).

## 19 Exceptions

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

20

21

# 1 Console.Write(System.Int64) Method

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

## 6 Summary

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

## 9 Parameters

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

10

## 11 Description

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

## 14 Exceptions

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

15

16

# 1 Console.WriteLine(System.UInt64) Method

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

## 6 Summary

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

## 9 Type Attributes:

- 10 • CLSCompliantAttribute(false)

## 11 Parameters

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

12

## 13 Description

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

16

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

## 19 Exceptions

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

20

21

1 **Console.Write(System.Object) Method**

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

6 **Summary**

7 Writes the text representation of a specified object to the  
8 System.Console.OutSystem.IO.TextWriter.

9 **Parameters**

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

10  
11 **Description**

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

14  
15 [Note: If *value* is null, no exception is thrown and nothing is written. Otherwise, the  
16 object's System.Object.ToString method is called to produce the string  
17 representation, and the resulting string is written to the output stream.]  
18  
19

20 **Exceptions**

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

21  
22

# 1 Console.WriteLine(System.String) Method

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

## 6 Summary

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

## 8 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.

## 9 10 Description

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

13  
14 [*Note:* If specified value is null, nothing is written to the output stream.]  
15  
16

## 17 Exceptions

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

18

19

# 1 Console.WriteLine(System.Boolean) Method

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

## 6 Summary

7 Writes the text representation of a System.Boolean followed by a line terminator to the  
8 System.Console.OutSystem.IO.TextWriter.

## 9 Parameters

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

10

## 11 Description

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

14

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

## 18 Exceptions

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

19

20

# 1 Console.WriteLine() Method

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

## 6 Summary

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

## 8 Description

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

## 12 Exceptions

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

13

14

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

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

## 8 Summary

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

## 11 Parameters

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

12

## 13 Description

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

17

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

19

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

21

22 ]

## 23 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 *arg.Length*.

1

2

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

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

## 10 Summary

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

## 13 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.

## 14 15 Description

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

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

21  
22 For more information on format strings, see the System.String class overview.

23  
24 ]

## 25 Exceptions

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

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

1

2

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

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

## 8 Summary

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

## 11 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.

12

## 13 Description

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

17

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

19

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

21

22 ]

## 23 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>format</i> is null.
<b>System.IO.IOException</b>	An I/O error occurred.

**System.FormatException**

The format specification in *format* 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

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

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

## 7 Summary

8 Writes a formatted string and a line terminator to the  
9 System.Console.OutSystem.IO.TextWriter.

## 10 Parameters

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

## 11 12 Description

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

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

18  
19 For more information on format strings, see the System.String class overview.

20  
21 ]

## 22 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

# 1 Console.WriteLine(System.String) Method

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

## 6 Summary

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

## 9 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.

10

## 11 Description

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

14  
15 The default line terminator is the value of the `System.Environment.NewLine` property.  
16 The line terminator can be set using the `System.IO.TextWriter.NewLine` property of  
17 the `System.Console.Out` stream.

## 18 Exceptions

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

19

20

# 1 Console.WriteLine(System.Object) Method

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

## 6 Summary

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

## 9 Parameters

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

## 10 11 Description

12 This version of `System.Console.WriteLine` is equivalent to  
13 `System.Console.Out.WriteLine( value.ToString() )`.  
14  
15 The default line terminator is the value of the `System.Environment.NewLine` property.  
16 The line terminator can be set using the `System.IO.TextWriter.NewLine` property of  
17 the `System.Console.Out` stream.

## 18 Exceptions

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

19  
20

# 1 Console.WriteLine(System.UInt64) Method

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

## 6 Summary

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

## 9 Type Attributes:

- 10 • `CLSCompliantAttribute(false)`

## 11 Parameters

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

## 13 Description

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

16  
17 This version of `System.Console.WriteLine` is equivalent to  
18 `System.Console.Out.WriteLine( value.ToString())`.

19  
20 The default line terminator is the value of the `System.Environment.NewLine` property.  
21 The line terminator can be set using the `System.IO.TextWriter.NewLine` property of  
22 the `System.Console.Out` stream.

## 23 Exceptions

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

# 1 Console.WriteLine(System.Int64) Method

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

## 6 Summary

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

## 9 Parameters

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

## 10 11 Description

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

14 The default line terminator is the value of the `System.Environment.NewLine` property.  
15 The line terminator can be set using the `System.IO.TextWriter.NewLine` property of  
16 the `System.Console.Out` stream.  
17

## 18 Exceptions

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

19  
20

# 1 Console.WriteLine(System.UInt32) Method

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

## 6 Summary

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

## 9 Type Attributes:

- 10 • `CLSCompliantAttribute(false)`

## 11 Parameters

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

## 13 Description

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

16  
17 This version of `System.Console.WriteLine` is equivalent to  
18 `System.Console.Out.WriteLine( value.ToString() )`.

19  
20 The default line terminator is the value of the `System.Environment.NewLine` property.  
21 The line terminator can be set using the `System.IO.TextWriter.NewLine` property of  
22 the `System.Console.Out` stream.

## 23 Exceptions

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

# 1 Console.WriteLine(System.Int32) Method

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

## 6 Summary

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

## 9 Parameters

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

10

## 11 Description

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

14

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

## 18 Exceptions

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

19

20

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

## 3 Console.WriteLine(System.Single) Method

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

### 8 Summary

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

### 11 Parameters

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

### 13 Description

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

16 The default line terminator is the value of the System.Environment.NewLine property.  
17 The line terminator can be set using the System.IO.TextWriter.NewLine property of  
18 the System.Console.Out stream.  
19

### 20 Exceptions

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

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

## 3 Console.WriteLine(System.Double) Method

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

### 8 Summary

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

### 11 Parameters

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

### 13 Description

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

16  
17 The default line terminator is the value of theSystem.Environment.NewLine property.  
18 The line terminator can be set using theSystem.IO.TextWriter.NewLine property of the  
19 System.Console.Out stream.

### 20 Exceptions

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

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

## 3 Console.WriteLine(System.Decimal) Method

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

### 8 Summary

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

### 11 Parameters

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

### 13 Description

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

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

### 20 Exceptions

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

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

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

## 8 Summary

9 Writes a subarray of characters followed by a line terminator to the  
10 System.Console.OutSystem.IO.TextWriter.

## 11 Parameters

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

12

## 13 Description

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

16

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

## 20 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.

**System.ArgumentNullException**

*buffer* is null.

1

2

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

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

## 6 Summary

7 Writes an array of characters followed by a line terminator to the  
8 System.Console.OutSystem.IO.TextWriter.

## 9 Parameters

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

10

## 11 Description

12 This version of System.Console.WriteLine is equivalent to  
13 System.Console.Out.WriteLine( *buffer* ).

14  
15 The default line terminator is the value of the System.Environment.NewLine property.  
16 The line terminator can be set using the System.IO.TextWriter.NewLine property of  
17 the System.Console.Out stream.

## 18 Exceptions

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

19

20

# 1 Console.WriteLine(System.Char) Method

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

## 7 Summary

8 Writes a character followed by a line terminator to the  
9 System.Console.OutSystem.IO.TextWriter.

## 10 Parameters

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

## 11 12 Description

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

15  
16 The default line terminator is the value of the System.Environment.NewLine property.  
17 The line terminator can be set using the System.IO.TextWriter.NewLine property of  
18 the System.Console.Out stream.

## 19 Exceptions

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

20  
21

# 1 Console.Error Property

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

## 7 Summary

8 Gets the system's standard error output stream.

## 9 Property Value

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

## 11 Description

12 This property is read-only.

13

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

15

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

18

19

20

# 1 Console.In Property

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

## 7 Summary

8 Gets the system's standard input stream.

## 9 Property Value

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

## 11 Description

12 This property is read-only.

13

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

15

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

18

19

20

# 1 Console.Out Property

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

## 7 Summary

8 Gets the system's standard output stream.

## 9 Property Value

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

## 11 Description

12 This property is read-only.

13  
14 This property can be redirected using the `System.Console.SetOut` method.

15  
16 [*Note:* If the application does not have a `System.Console`, `System.Console.Out`  
17 behaves like `System.IO.TextWriter.Null`.]

18  
19

20