

1 System.Console Class

2
3

```
4 [ILASM]  
5 .class public sealed Console extends System.Object  
6 [C#]  
7 public sealed class Console
```

8 Assembly Info:

- 9 • Name: mscorlib
- 10 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 11 • Version: 1.0.x.x
- 12 • Attributes:
 - 13 ○ CLSCompliantAttribute(true)

14 Summary

15

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

18 Inherits From: System.Object

19

20 Library: BCL

21

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

23

24 Description

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

29

30 The standard input, output, and error streams are represented by
31 properties, and are automatically associated with the console when the
32 application starts. Applications can redirect these properties to other
33 streams; for example, streams associated with files instead of the
34 console. [Note: For additional information see the
35 **System.Console.SetIn**, **System.Console.SetOut**, and
36 **System.Console.SetError** methods.] By default, the read methods in
37 this class use the standard input stream and the write methods use
38 the standard output stream.

39

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

1 them to strings.
2
3 This class uses synchronized **System.IO.TextReader** and
4 **System.IO.TextWriter** instances. Multiple threads can concurrently
5 read from and/or write to an instance of this type.

6 **Example** 7

8 The following example demonstrates the use of basic
9 **System.Console** input and output functions. The program waits for
10 the user to enter a name.

```
11          [C#]  
12  
13  
14          using System;  
15  
16          public class ConsoleTest {  
17              public static void Main() {  
18                  Console.Write("Hello ");  
19                  Console.WriteLine("World!");  
20                  Console.Write("What is your name: ");  
21                  string name = Console.ReadLine();  
22                  Console.Write("Hello, ");  
23                  Console.Write(name);  
24                  Console.WriteLine("!");  
25              }  
26          }
```

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

28
29 Hello World!

30
31
32 What is your name: Fred

33
34
35 Hello, Fred!
36

37

1 Console.OpenStandardError() Method

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

7 Summary

8 Returns the standard error stream.

9 Return Value

10

11 A new synchronized **System.IO.Stream** object that writes to the
12 console.

13

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

11
12

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

13
14
15

14 Return Value

16 A new synchronized **System.IO.Stream** object that writes to the
17 console.

18 Description

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

24

1 Console.OpenStandardInput() Method

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

7 Summary

8 Returns the standard input stream.

9 Return Value

10

11 A new synchronized **System.IO.Stream** object that reads from the
12 console.

13

1 Console.OpenStandardInput(System.Int32) Method

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

8 Summary

9 Returns the standard input stream.

10 Parameters

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

14 Return Value

16 A new synchronized **System.IO.Stream** object that reads from the
17 console.

18 Description

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

24

1 Console.OpenStandardOutput() Method

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

7 Summary

8 Returns the standard output stream.

9 Return Value

10

11 A new synchronized **System.IO.Stream** object that writes to the
12 console.

13

1 Console.OpenStandardOutput(System.Int 2 32) 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
10 buffer for the stream is specified.

11 Parameters

12
13

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

14
15
16

15 Return Value

17 A new synchronized **System.IO.Stream** object that writes to the
18 console.

19 Description

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

25

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

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

12 Description

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

17 Exceptions

18

19

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

20

21

22

1 Console.ReadLine() Method

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

6 Summary

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

9 Return Value

10

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

13 Description

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

18 Exceptions

19

20

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

21

22

23

1 Console.SetError(System.IO.TextWriter) 2 Method

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

8 Summary

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

11 Parameters

12
13

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

14
15

15 Description

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

19
20
21

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

22 Exceptions

23
24

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

25
26
27
28

26 Permissions

Permission	Description
System.Security.Permissions.SecurityPermissionFlag	Requires System.Security.Permissions.SecurityPermissionFlag UnmanagedCode permission.

1
2
3

1 Console.SetIn(System.IO.TextReader)

2 Method

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

8 Summary

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

11 Parameters

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

15 Description

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

19
20 [Note: By default, the **System.Console.In** property is set to the
21 system's standard input stream.]

22 Exceptions

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

26 Permissions

Permission	Description
System.Security.Permissions.SecurityPermissionFlag	Requires permission to execute unmanaged code. See System.Security.Permissions.SecurityPermissionFlag.UnmanagedCode .

1 Console.SetOut(System.IO.TextWriter) 2 Method

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

8 Summary

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

11 Parameters

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

15 Description

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

19
20 [Note: By default, the **System.Console.Out** property is set to the
21 system's standard output stream.]

22 Exceptions

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

26 Permissions

Permission	Description
System.Security.Permissions.SecurityPermissionFlag	Requires permission to execute unmanaged code. See System.Security.Permissions.SecurityPermissionFlag.UnmanagedCode .

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

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

8 Summary

9 Writes a formatted string to the **System.Console.Out**
10 **System.IO.TextWriter**.

11 Parameters

12
13

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

14
15

Description

16 [Note: If the specified object is not referenced in *format*, it is ignored.
17
18 For more information on format strings, see the **System.String** class
19 overview.]

20 Exceptions

21
22

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

23
24
25

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

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

9 Summary

10 Writes a formatted string to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Parameters

13
14

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.

15
16

Description

17 [Note: If a specified object is not referenced in *format*, it is ignored.
18
19 For more information on format strings, see the **System.String** class
20 overview.]

21 Exceptions

22
23

Exception	Condition
System.ArgumentNullException	<i>format</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	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
3

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

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

10 Summary

11 Writes a formatted string to the **System.Console.Out**
12 **System.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.

16 17 Description

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

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

23 Exceptions

Exception	Condition
System.ArgumentNullException	<i>format</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	The format specification in <i>format</i> is invalid. -or-

- 1
- 2
- 3

	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 Console.Write(System.String, 2 System.Object[]) Method

```
3 [ILASM]  
4 .method public hidebysig static void Write(string format,  
5 class System.Object[] arg)  
  
6 [C#]  
7 public static void Write(string format, params object[]  
8 arg)
```

9 Summary

10 Writes a formatted string to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Parameters

13
14

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.

15

16 Description

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

18

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

21 Exceptions

22

23

Exception	Condition
System.ArgumentNullException	<i>format</i> or <i>arg</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	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</i> .Length.

1
2
3

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.Out** **System.IO.TextWriter**.

9 Parameters

10
11

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

12

13 Description

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

16 Exceptions

17
18

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

19
20
21

1 Console.Write(System.Char) Method

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

7 Summary

8 Writes a character to the **System.Console.Out**
9 **System.IO.TextWriter**.

10 Parameters

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

13 Exceptions

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

17
18
19

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

```
2 [ILASM]  
3 .method public hidebysig static void Write(class  
4 System.Char[] buffer)  
  
5 [C#]  
6 public static void Write(char[] buffer)
```

7 Summary

8 Writes a character array to the **System.Console.Out**
9 **System.IO.TextWriter**.

10 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.

13 Description

15 This version of **System.Console.Write** is equivalent to
16 **System.Console.Out.Write(buffer)**.

17 Exceptions

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

20
21
22

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

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

9 Summary

10 Writes a subarray of characters to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Parameters

13
14

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

15

16 Description

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

19 Exceptions

20
21

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

22
23
24

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

3 Console.Write(System.Double) Method

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

8 Summary

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

11 Parameters

12
13

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

14

15 Description

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

18 Exceptions

19
20

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

21

22

23

1 **The following member must be implemented if the ExtendedNumerics library is**
2 **present in 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.Out** **System.IO.TextWriter**.

11 Parameters

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

14 Description

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

18 Exceptions

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

21
22
23

1 **The following member must be implemented if the ExtendedNumerics library is**
2 **present in 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.Out** **System.IO.TextWriter**.

11 Parameters

12
13

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

14

15 Description

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

18 Exceptions

19
20

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

21

22

23

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

10
11

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

12
13

13 Description

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

16 Exceptions

17
18

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

19
20
21

1 Console.WriteLine(System.UInt32) Method

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

7 Summary

8 Writes the text representation of a specified **System.UInt32** to the
9 **System.Console.Out** **System.IO.TextWriter**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13
14

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

15
16

16 Description

17 This member is not CLS-compliant. For a CLS-compliant alternative,
18 use **System.Console.WriteLine(System.Int64)**.

19
20
21

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

22 Exceptions

23
24

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

25
26
27

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

10
11

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

12

13 Description

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

16 Exceptions

17
18

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

19
20
21

1 Console.Write(System.UInt64) Method

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

7 Summary

8 Writes the text representation of a specified **System.UInt64** to the
9 **System.Console.Out** **System.IO.TextWriter**.

10 Type Attributes:

- 11 • CLSCompliantAttribute(false)

12 Parameters

13 Parameter	14 Description
<i>value</i>	The System.UInt64 to write.

16 Description

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

19
20 This version of **System.Console.Write** is equivalent to
21 **System.Console.Out.Write(value.ToString())**.

22 Exceptions

23 Exception	24 Condition
System.IO.IOException	An I/O error occurred.

25
26
27

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.Out** **System.IO.TextWriter**.

9 Parameters

10
11

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

12
13

13 Description

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

16
17
18
19
20

[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.]

21 Exceptions

22
23

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

24
25
26

1 Console.Write(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.Out**
8 **System.IO.TextWriter**.

9 Parameters

10
11

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

12
13

Description

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

16
17 [Note: If specified value is **null**, nothing is written to the output
18 stream.]

19 Exceptions

20
21

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

22
23
24

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**
8 **System.IO.TextWriter**.

9 Description

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

14 Exceptions

15
16

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

17
18
19

1 Console.WriteLine(System.Boolean)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static void WriteLine(bool value)  
5  
6 [C#]  
7 public static void WriteLine(bool value)
```

7 Summary

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

10 Parameters

11
12

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

13
14

14 Description

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

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

22 Exceptions

23
24

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

25
26
27

1 Console.WriteLine(System.Char) Method

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

7 Summary

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

10 Parameters

11
12

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

13
14

14 Description

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

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

22 Exceptions

23
24

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

25
26
27

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

```
2 [ILASM]  
3 .method public hidebysig static void WriteLine(class  
4 System.Char[] buffer)  
  
5 [C#]  
6 public static void WriteLine(char[] buffer)
```

7 Summary

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

10 Parameters

Parameter	Description
<i>buffer</i>	The Unicode character array to write.

14 Description

15 This version of **System.Console.WriteLine** is equivalent to
16 **System.Console.Out.WriteLine(*buffer*)**.

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

22 Exceptions

Exception	Condition
System.IO.IOException	An I/O error occurred.
System.ArgumentException	<i>buffer</i> is null .

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

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

9 Summary

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

12 Parameters

13
14

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.

15
16

16 Description

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

19
20
21
22
23

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.

24 Exceptions

25
26

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

1
2
3

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

3 Console.WriteLine(System.Decimal) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static void WriteLine(decimal  
7 value)  
8 [C#]  
9 public static void WriteLine(decimal value)
```

10 Summary

11 Writes the text representation of a specified **System.Decimal**
12 followed by a line terminator to the **System.Console.Out**
13 **System.IO.TextWriter**.

14 Parameters

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

18 Description

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

21 The default line terminator is the value of the
22 **System.Environment.NewLine** property. The line terminator can be
23 set using the **System.IO.TextWriter.NewLine** property of the
24 **System.Console.Out** stream.
25

26 Exceptions

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

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

3 Console.WriteLine(System.Double) 4 Method

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

10 Summary

11 Writes the text representation of a specified **System.Double** followed
12 by a line terminator to the **System.Console.Out**
13 **System.IO.TextWriter**.

14 Parameters

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

18 Description

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

21 The default line terminator is the value of
22 the **System.Environment.NewLine** property. The line terminator can
23 be set using the **System.IO.TextWriter.NewLine** property of the
24 **System.Console.Out** stream.
25

26 Exceptions

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

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

3 Console.WriteLine(System.Single) Method

4 [ILASM]
5 .method public hidebysig static void WriteLine(float32
6 value)

7 [C#]
8 public static void WriteLine(float value)

9 Summary

10 Writes the text representation of a specified **System.Single** followed
11 by a line terminator to the **System.Console.Out**
12 **System.IO.TextWriter**.

13 Parameters

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

16

17 Description

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

20

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

25 Exceptions

26

27

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

28

29

30

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
8 by a line terminator to the **System.Console.Out**
9 **System.IO.TextWriter**.

10 Parameters

11
12

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

13

14 Description

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

17

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

22 Exceptions

23

24

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

25

26

27

1 Console.WriteLine(System.UInt32)

2 Method

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

8 Summary

9 Writes the text representation of a specified **System.UInt32** followed
10 by a line terminator to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15
16

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

17

18 Description

19 This member is not CLS-compliant. For a CLS-compliant alternative,
20 use **System.Console.WriteLine(System.Int64)**.

21

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

24

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

29 Exceptions

30

31

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

32

33

34

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
8 by a line terminator to the **System.Console.Out**
9 **System.IO.TextWriter**.

10 Parameters

11
12

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

13

14 Description

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

17

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

22 Exceptions

23

24

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

25

26

27

1 Console.WriteLine(System.UInt64)

2 Method

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

8 Summary

9 Writes the text representation of a specified **System.UInt64** followed
10 by a line terminator to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Type Attributes:

- 13 • CLSCompliantAttribute(false)

14 Parameters

15
16

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

17

18 Description

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

21

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

24

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

29 Exceptions

30

31

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

32

33

34

1 Console.WriteLine(System.Object) Method

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

7 Summary

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

10 Parameters

11
12

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

13
14

Description

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

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

22 Exceptions

23
24

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

25
26
27

1 Console.WriteLine(System.String) Method

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

7 Summary

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

10 Parameters

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

13 Description

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

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

22 Exceptions

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

25
26
27

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

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

8 Summary

9 Writes a formatted string and a line terminator to the
10 **System.Console.Out** **System.IO.TextWriter**.

11 Parameters

12
13

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

14
15

15 Description

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

20

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

22

23 For more information on format strings, see the **System.String** class
24 overview.]

25 Exceptions

26
27

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

- 1
- 2
- 3

	equal to the number of provided objects to be formatted (1).
--	--

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

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

9 Summary

10 Writes a formatted string and a new line to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 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.

16 Description

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

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

22 For more information on format strings, see the **System.String** class
23 overview.]
24
25

26 Exceptions

Exception	Condition
System.ArgumentNullException	<i>format</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	The format specification in <i>format</i> is invalid. -or-

- 1
- 2
- 3

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

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

10 Summary

11 Writes a formatted string and a new line to the **System.Console.Out**
12 **System.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.

16 Description

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

22
23 [Note: If a specified object is not referenced in *format*, it is ignored.

24
25 For more information on format strings, see the **System.String** class
26 overview.]

27 Exceptions

Exception	Condition
System.ArgumentNullException	<i>format</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	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 the number of provided objects to be formatted (3).</p>
--	---

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

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

9 Summary

10 Writes a formatted string and a new line to the **System.Console.Out**
11 **System.IO.TextWriter**.

12 Parameters

13
14

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.

15
16

16 Description

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

21
22
23
24
25

[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.]

26 Exceptions

27
28

Exception	Condition
System.ArgumentNullException	<i>format</i> or <i>arg</i> is null .
System.IO.IOException	An I/O error occurred.
System.FormatException	The format specification in <i>format</i> is invalid. -or-

1
2
3

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

1 Console.Error Property

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

8 Summary

9 Gets the system's standard error output stream.

10 Property Value

11

12 A synchronized **System.IO.TextWriter** object where error output is
13 sent.

14 Description

15 This property is read-only.

16

17 This property can be redirected using the **System.Console.SetError**
18 method.

19

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

22

1 Console.In Property

```
2 [ILASM]
3 .property class System.IO.TextReader In { public hideby sig
4 static 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

11 A synchronized **System.IO.TextReader** object from which user input
12 is received.

13 Description

14 This property is read-only.

15

16 This property can be redirected using the **System.Console.SetIn**
17 method.

18

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

21

1 Console.Out Property

```
2 [ILASM]  
3 .property class System.IO.TextWriter Out { public hidebysig  
4 static 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

11 A synchronized **System.IO.TextWriter** object where normal output is
12 sent.

13 Description

14 This property is read-only.

15

16 This property can be redirected using the **System.Console.SetOut**
17 method.

18

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

21