

1 System.IO.TextWriter Class

2
3

```
4 [ILASM]  
5 .class public abstract serializable TextWriter extends  
6 System.MarshalByRefObject implements System.IDisposable  
  
7 [C#]  
8 public abstract class TextWriter: MarshalByRefObject,  
9 IDisposable
```

10 Assembly Info:

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

16 Implements:

- 17 • System.IDisposable

18 Summary

19

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

21 Inherits From: System.MarshalByRefObject

22

23 Library: BCL

24

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

27

28 Description

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

31

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

34

1 TextWriter() Constructor

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

6 Summary

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

9

1 TextWriter(System.IFormatProvider)

2 Constructor

```
3 [ILASM]  
4 family rtspecialname specialname instance void .ctor(class  
5 System.IFormatProvider formatProvider)  
6 [C#]  
7 protected TextWriter(IFormatProvider formatProvider)
```

8 Summary

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

11 Parameters

12
13

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

14
15
16

1 TextWriter.Null Field

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

7 Summary

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

9 Description

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

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

16

1 **TextWriter.Close()** Method

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

6 **Summary**

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

9 **Description**

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

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

15 **Behaviors**

16 As described above.

17

1 `TextWriter.Dispose(System.Boolean)`

2 Method

```
3 [ILASM]  
4 .method family hidebysig virtual void Dispose(bool  
5 disposing)  
6  
7 [C#]  
protected virtual void Dispose(bool disposing)
```

8 Summary

9 Releases the unmanaged resources used by the
10 **System.IO.TextWriter** and optionally releases the managed
11 resources.

12 Parameters

13
14

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

15
16

Description

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

21
22
23
24
25
26

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

27

1 TextWriter.Flush() Method

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

6 Summary

7 Writes any buffered data to the underlying device and clears all buffers
8 for the current **System.IO.TextWriter** instance.

9 Behaviors

10 As described above.

11 Default

12 This method is not implemented in this abstract class.

13

1 `TextWriter.Synchronized(System.IO.Text` 2 `Writer)` Method

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

8 Summary

9 Creates a thread-safe wrapper around the specified
10 **System.IO.TextWriter** instance.

11 Parameters

12
13

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

14
15
16

15 Return Value

17 A thread-safe **System.IO.TextWriter**.

18 Description

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

22 Exceptions

23
24

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

25
26
27

1 **TextWriter.System.IDisposable.Dispose()**

2 **Method**

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

8 **Summary**

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

11

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

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

9 Summary

10 Writes a formatted string to the text stream.

11 Parameters

12
13

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg</i>	The array of objects referenced from the format string.

14

15 Description

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

19

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

22

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

26 Behaviors

27 As described above.

28 Exceptions

29

30

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

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

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

10 Summary

11 Writes out a formatted string to the text stream.

12 Parameters

13
14

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.
<i>arg2</i>	The third object referenced in the format string.

15

16 Description

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

20

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

23

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

27 Behaviors

28 As described above.

29 Exceptions

30

31

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

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

- 1
- 2
- 3

1 `TextWriter.Write(System.String,` 2 `System.Object, System.Object)` Method

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

9 Summary

10 Writes out a formatted string to the text stream.

11 Parameters

12
13

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.

14

15 Description

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

19

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

22

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

26 Behaviors

27 As described above.

28 Exceptions

29

30

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

	-or- An object referenced in the <i>format</i> string 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 **TextWriter.Write(System.String,** 2 **System.Object) Method**

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

8 **Summary**

9 Writes out the format string to the text stream.

10 **Parameters**

11
12

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

13
14

14 **Description**

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

18
19
20
21

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

22
23
24

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

25 **Behaviors**

26 As described above.

27 **Exceptions**

28
29

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

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

1
2
3

1 `TextWriter.Write(System.Object)` Method

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

6 **Summary**

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

8 **Parameters**

9
10

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

11
12

Description

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

15 **Behaviors**

16 As described above.

17 **Default**

18 The default implementation uses the
19 **System.IO.TextWriter.Write(System.String)** method.

20 **Exceptions**

21
22

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

23
24
25

1 **TextWriter.Write(System.String) Method**

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

6 **Summary**

7 Writes a specified **System.String** to the text stream.

8 **Parameters**

9
10

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

11
12

Description

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

15
16
17
18

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

19 **Behaviors**

20 As described above.

21 **Exceptions**

22
23

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

24
25
26

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

3 **TextWriter.Write(System.Decimal)** 4 **Method**

```
5 [ILASM]  
6 .method public hidebysig virtual void Write(decimal value)  
7 [C#]  
8 public virtual void Write(decimal value)
```

9 **Summary**

10 Writes the text representation of a specified **System.Decimal** to the
11 text stream.

12 **Parameters**

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

16 **Description**

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

19 **Behaviors**

20 As described above.

21 **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 **TextWriter.Write(System.Double) Method**

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

8 **Summary**

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

11 **Parameters**

12
13

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

14

15 **Description**

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

18 **Behaviors**

19 As described above.

20 **Exceptions**

21
22

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

23

24

25

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

3 **TextWriter.Write(System.Single) Method**

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

8 **Summary**

9 Writes the text representation of a specified **System.Single** to the
10 text stream.

11 **Parameters**

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

14 **Description**

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

18 **Behaviors**

19 As described above.

20 **Exceptions**

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

23
24
25

1 **TextWriter.Write(System.UInt64) Method**

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

7 **Summary**

8 Writes the text representation of a specified **System.UInt64** to the
9 text stream.

10 **Type Attributes:**

- 11 • CLSCompliantAttribute(false)

12 **Parameters**

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

15 **Description**

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

18 This version of **System.IO.TextWriter.Write** is equivalent to
19 **System.IO.TextWriter.Write(*value*.ToString ())**.
20
21

22 **Behaviors**

23 As described above.

24 **Exceptions**

25 Exception	26 Condition
System.IO.IOException	An I/O error occurred.

27
28
29

1 **TextWriter.Write(System.Int64) Method**

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

6 **Summary**

7 Writes the text representation of a specified **System.Int64** to the text
8 stream.

9 **Parameters**

10
11

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

12
13

13 **Description**

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

16 **Behaviors**

17 As described above.

18 **Exceptions**

19
20

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

21
22
23

1 `TextWriter.Write(System.UInt32)` Method

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

7 **Summary**

8 Writes the text representation of a **System.UInt32** to the text
9 stream.

10 **Type Attributes:**

- 11 • `CLSCompliantAttribute(false)`

12 **Parameters**

13 Parameter	14 Description
<i>value</i>	The System.UInt32 to write.

15 **Description**

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

18 This version of **System.IO.TextWriter.Write** is equivalent to
19 **System.IO.TextWriter.Write**(*value*.ToString()).
20
21

22 **Behaviors**

23 As described above.

24 **Exceptions**

25 Exception	26 Condition
System.IO.IOException	An I/O error occurred.

27
28
29

1 **TextWriter.Write(System.Int32) Method**

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

6 **Summary**

7 Writes the text representation of a specified **System.Int32** to the text
8 stream.

9 **Parameters**

10
11

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

12

13 **Description**

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

16 **Behaviors**

17 As described above.

18 **Exceptions**

19
20

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

21
22
23

1 **TextWriter.Write(System.Boolean)**

2 **Method**

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

7 **Summary**

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

10 **Parameters**

11
12

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

13

14 **Description**

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

17 **Behaviors**

18 As described above.

19 **Exceptions**

20
21

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

22
23
24

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

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

9 Summary

10 Writes a subarray of characters to the text stream.

11 Parameters

12
13

Parameter	Description
<i>buffer</i>	The character array from which characters are read.
<i>index</i>	Starting index in the buffer.
<i>count</i>	The number of characters to write.

14

15 Description

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

19 Behaviors

20 As described above.

21 Default

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

24 Exceptions

25
26

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentException	$(index + count) > buffer.Length$.
System.ArgumentOutOfRangeException	$index < 0$ or $count < 0$.

1
2
3

System.IO.IOException	An I/O error occurred.
------------------------------	------------------------

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

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

7 **Summary**

8 Writes a character array to the text stream.

9 **Parameters**

10
11

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

12
13

Description

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

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

20 **Behaviors**

21 As described above.

22 **Exceptions**

23
24

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

25
26
27

1 TextWriter.Write(System.Char) Method

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

7 Summary

8 Writes a character to the text stream.

9 Parameters

10
11

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

12
13

14 Behaviors

15 As described above.

16 Default

17 This method is not implemented in this abstract class.

18 Exceptions

19
20

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

21
22
23

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

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

9 Summary

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

12 Parameters

13
14

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg</i>	The array of objects referenced from the format string.

15

16 Description

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

20

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

23

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

27 Behaviors

28 As described above.

29 Exceptions

30

31

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

1
2
3

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

1 `TextWriter.WriteLine(System.String,` 2 `System.Object, System.Object,` 3 `System.Object)` Method

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

10 Summary

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

13 Parameters

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.
<i>arg2</i>	The third object referenced in the format string.

16 17 Description

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

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

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

28 Behaviors

29 As described above.

30 Exceptions

31
32

Exception	Condition
System.ArgumentNullException	<i>format</i> is null . -or- An object referenced in the <i>format</i> string 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 (3).

1
2
3

1 `TextWriter.WriteLine(System.String,` 2 `System.Object, System.Object)` Method

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

9 Summary

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

12 Parameters

13
14

Parameter	Description
<i>format</i>	A System.String containing the format string.
<i>arg0</i>	The first object referenced in the format string.
<i>arg1</i>	The second object referenced in the format string.

15
16

Description

17 This version of **System.IO.TextWriter.WriteLine** is equivalent to
18 **System.IO.TextWriter.WriteLine(System.String.Format** (*format*,
19 *arg0*, *arg1*)).
20

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

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

27 Behaviors

28 As described above.

29 Exceptions

30
31

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

	-or- An object referenced in the <i>format</i> string 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 `TextWriter.WriteLine(System.String, System.Object)` Method

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

8 Summary

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

11 Parameters

12
13

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

14
15

15 Description

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

19
20
21

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

22
23
24
25

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

26 Behaviors

27 As described above.

28 Exceptions

29
30

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

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

1
2
3

1 **TextWriter.WriteLine(System.Object)**

2 **Method**

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

8 **Summary**

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

11 **Parameters**

12
13

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

14
15

15 **Description**

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

18 **Behaviors**

19 As described above.

20 **Exceptions**

21
22

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

23
24
25

1 **TextWriter.WriteLine(System.String)**

2 **Method**

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

8 **Summary**

9 Writes a specified **System.String** followed by a line terminator to the
10 text stream.

11 **Parameters**

12
13

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

14
15

15 **Description**

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

18
19 The line terminator string is defined by the
20 **System.IO.TextWriter.NewLine** property.

21
22 [*Note:* This method does not search the specified **System.String** for
23 individual newline characters (hexadecimal 0x000a) and replace them
24 with **System.IO.TextWriter.NewLine**.]

25 **Behaviors**

26 As described above.

27 **Exceptions**

28
29

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

30
31
32

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

3 **TextWriter.WriteLine(System.Decimal)** 4 **Method**

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

10 **Summary**

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

13 **Parameters**

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

17 **Description**

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

20 **Behaviors**

21 As described above.

22 **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 **TextWriter.WriteLine(System.Double)** 4 **Method**

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

10 **Summary**

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

13 **Parameters**

14
15

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

16
17

17 **Description**

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

20 **Behaviors**

21 As described above.

22 **Exceptions**

23
24

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

25
26
27

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

3 **TextWriter.WriteLine(System.Single)** 4 **Method**

```
5 [ILASM]  
6 .method public hidebysig virtual void WriteLine(float32  
7 value)  
8 [C#]  
9 public virtual void WriteLine(float value)
```

10 **Summary**

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

13 **Parameters**

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

16 **Description**

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

19 **Behaviors**

20 As described above.

21 **Exceptions**

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

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

2 **Method**

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

8 **Summary**

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

11 **Type Attributes:**

- 12 • CLSCompliantAttribute(false)

13 **Parameters**

14
15

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

16
17

17 **Description**

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

20
21
22

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

23 **Behaviors**

24 As described above.

25 **Exceptions**

26
27

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

28
29
30

1 **TextWriter.WriteLine(System.Int64)**

2 **Method**

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

8 **Summary**

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

11 **Parameters**

12
13

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

14

15 **Description**

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

18 **Behaviors**

19 As described above.

20 **Exceptions**

21
22

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

23

24

25

1 `TextWriter.WriteLine(System.UInt32)`

2 Method

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

8 Summary

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

11 Type Attributes:

- 12 • `CLSCompliantAttribute(false)`

13 Parameters

14
15

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

16
17

Description

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

20
21
22

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

23 Behaviors

24 As described above.

25 Exceptions

26
27

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

28
29
30

1 `TextWriter.WriteLine(System.Int32)`

2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual void WriteLine(int32  
5 value)  
6  
7 [C#]  
8 public virtual void WriteLine(int value)
```

8 Summary

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

11 Parameters

12
13

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

14

15 Description

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

18 Behaviors

19 As described above.

20 Exceptions

21
22

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

23

24

25

1 **TextWriter.WriteLine(System.Boolean)**

2 **Method**

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

7 **Summary**

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

10 **Parameters**

11
12

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

13

14 **Description**

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

17 **Behaviors**

18 As described above.

19 **Exceptions**

20
21

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

22
23
24

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

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

9 Summary

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

12 Parameters

13
14

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

15
16

16 Description

17 This version of **System.IO.TextWriter.WriteLine** is equivalent to
18 **System.IO.TextWriter.Write**(*value*, *index*, *count*) followed by
19 **System.IO.TextWriter.WriteLine**().
20

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

24 Behaviors

25 As described above.

26 Exceptions

27
28

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentException	(<i>index</i> + <i>count</i>) > <i>buffer</i> .

1
2
3

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

1 `TextWriter.WriteLine(System.Char[])`

2 Method

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

8 Summary

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

11 Parameters

12
13

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

14
15

15 Description

16 This version of **System.IO.TextWriter.WriteLine** is equivalent to
17 **System.IO.TextWriter.Write(*value*)** followed by
18 **System.IO.TextWriter.WriteLine()**.

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

23 Behaviors

24 As described above.

25 Exceptions

26
27

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

28
29
30

1 **TextWriter.WriteLine(System.Char)**

2 **Method**

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

8 **Summary**

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

11 **Parameters**

12
13

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

14

15 **Description**

16 This version of **System.IO.TextWriter.WriteLine** is equivalent to
17 **System.IO.TextWriter.Write(*value*)** followed by
18 **System.IO.TextWriter.WriteLine()**.

19 **Behaviors**

20 As described above.

21 **Exceptions**

22
23

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

24
25
26

1 `TextWriter.WriteLine()` Method

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

6 **Summary**

7 Writes a line terminator to the text stream.

8 **Description**

9 This version of **System.IO.TextWriter.WriteLine** is equivalent to
10 **System.IO.TextWriter.Write(System.IO.TextWriter.NewLine.To**
11 **CharArray())**.

12 **Behaviors**

13 As described above.

14 **Exceptions**

15
16

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

17
18
19

1 TextWriter.Encoding Property

```
2 [ILASM]
3 .property class System.Text.Encoding Encoding { public
4 hidebysig virtual abstract specialname class
5 System.Text.Encoding get_Encoding() }
6
7 [C#]
8 public abstract Encoding Encoding { get; }
```

8 Summary

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

10 Property Value

11

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

13 Behaviors

14 As described above.

15 Usage

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

20

1 TextWriter.FormatProvider Property

```
2 [ILASM]
3 .property class System.IFormatProvider FormatProvider {
4 public hidebysig virtual specialname class
5 System.IFormatProvider get_FormatProvider() }
6
7 [C#]
8 public virtual IFormatProvider FormatProvider { get; }
```

8 Summary

9 Gets an object that provides formatting for the current instance

10 Property Value

11

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

14 Behaviors

15 As described above.

16

1 **TextWriter.NewLine** Property

```
2 [ILASM]  
3 .property string NewLine { public hidebysig virtual  
4 specialname string get_NewLine() public hidebysig virtual  
5 specialname void set_NewLine(string value) }  
6 [C#]  
7 public virtual string NewLine { get; set; }
```

8 **Summary**

9 Gets or sets the line terminator string used by the current
10 **System.IO.TextWriter**.

11 **Property Value**

13 A **System.String** containing the line terminator string for the current
14 **System.IO.TextWriter**.

15 **Description**

16 The default line terminator string is **System.Environment.NewLine**.

17
18 The line terminator string is written to the text stream whenever one
19 of the **System.IO.TextWriter.WriteLine** methods is called. In order
20 for text written by the **System.IO.TextWriter** to be readable by a
21 **System.IO.TextReader**, only "\n" or
22 **System.Environment.NewLine** should be used as terminator
23 strings. If **System.IO.TextWriter.NewLine** is set to **null**, the default
24 newline character is used instead.

25 **Behaviors**

26 As described above.

27