

System.IO.StringWriter Class

```
[ILASM]
.class public serializable StringWriter extends
System.IO.TextWriter

[C#]
public class StringWriter: TextWriter
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

- **System.IDisposable**

Summary

Implements a **System.IO.TextWriter** that writes information to a string.

Inherits From: System.IO.TextWriter

Library: BCL

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

1 StreamWriter() Constructor

```
2 [ILASM]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public StreamWriter()
```

6 Summary

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

9 Description

10 A new **System.Text.StringBuilder** is automatically created and
11 associated with the new **System.IO.StringWriter** instance.

12

1 StreamWriter(System.IFormatProvider)

2 Constructor

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

8 Summary

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

11 Parameters

12
13

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

14
15
16

1 StreamWriter(System.Text.StringBuilder)

2 Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.Text.StringBuilder sb)  
  
6 [C#]  
7 public StreamWriter(StringBuilder sb)
```

8 Summary

9 Constructs and initializes a new instance of the
10 **System.IO.StringWriter** class that writes to the specified
11 **System.Text.StringBuilder**.

12 Parameters

13
14

Parameter	Description
<i>sb</i>	The System.Text.StringBuilder to write to.

15
16
17
18

Exceptions

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

19
20
21

1 StreamWriter(System.Text.StringBuilder, 2 System.IFormatProvider) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.Text.StringBuilder sb, class System.IFormatProvider  
6 formatProvider)  
  
7 [C#]  
8 public StreamWriter(StringBuilder sb, IFormatProvider  
9 formatProvider)
```

10 Summary

11 Constructs and initializes a new instance of the
12 **System.IO.StreamWriter** class that writes to the specified
13 **System.Text.StringBuilder** and has the specified format provider.

14 Parameters

Parameter	Description
<i>sb</i>	The System.Text.StringBuilder to write to.
<i>formatProvider</i>	A System.IFormatProvider object that defines formatting.

17 Exceptions

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

21
22
23

1 StreamWriter.Close() Method

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

6 Summary

7 Closes the current **System.IO.StringWriter** instance.

8 Description

9 [Note: This version of **System.IO.StringWriter.Close** is equivalent to
10 **System.IO.StringWriter.Dispose(true)**.
11

12 This method overrides **System.IO.Stream.Close.**]
13

1 `StringWriter.Dispose(System.Boolean)`

2 Method

3 `[ILASM]`
4 `.method family hidebysig virtual void Dispose(bool`
5 `disposing)`

6 `[C#]`
7 `protected override void Dispose(bool disposing)`

8 Summary

9 Releases the unmanaged resources used by the
10 **System.IO.StringWriter** 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.StringWriter** references. This method invokes the
20 **Dispose()** method of each referenced object.

21

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

27

1 **StringWriter.GetStringBuilder()** Method

```
2 [ILASM]  
3 .method public hidebysig virtual class  
4 System.Text.StringBuilder GetStringBuilder()  
  
5 [C#]  
6 public virtual StringBuilder GetStringBuilder()
```

7 **Summary**

8 Returns the underlying **System.Text.StringBuilder**.

9 **Return Value**

10

11 The underlying **System.Text.StringBuilder**.

12 **Description**

13 This method returns either the **System.Text.StringBuilder** that was
14 passed to the constructor, or the **System.Text.StringBuilder** that
15 was automatically created.

16 **Behaviors**

17 As described above.

18

1 StreamWriter.ToString() Method

```
2 [ILASM]  
3 .method public hidebysig virtual string ToString()  
4 [C#]  
5 public override string ToString()
```

6 Summary

7 Returns a string containing the characters written to the current
8 **System.IO.StringWriter** instance so far.

9 Return Value

10

11 The **System.String** containing the characters written to the current
12 **System.IO.StringWriter** instance.

13 Description

14 [*Note:* This method overrides **System.Object.ToString.**]

15

1 StreamWriter.Write(System.String) Method

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

6 Summary

7 Writes a string to the current instance of the
8 **System.IO.StringWriter**.

9 Parameters

10
11

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

12

13 Description

14 [Note: This method overrides **System.IO.TextWriter.Write**.]

15 Exceptions

16
17

Exception	Condition
System.ObjectDisposedException	The writer is closed.

18

19

20

1 `StringWriter.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 override void Write(char[] buffer, int index, int  
8 count)
```

9 Summary

10 Writes the specified region of a character array to this instance of the
11 **System.IO.StringWriter**.

12 Parameters

13
14

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

15
16

Description

17 [Note: This method overrides **System.IO.TextWriter.Write**.]

18 Exceptions

19
20

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentOutOfRangeException	<i>index</i> < 0 -or- <i>count</i> < zero.
System.ArgumentException	(<i>index</i> + <i>count</i>) > <i>buffer</i> .Length.
System.ObjectDisposedException	The writer is closed.

21
22
23

1 StreamWriter.Write(System.Char) Method

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

7 Summary

8 Writes a character to the current instance of the
9 **System.IO.StringWriter**.

10 Parameters

11
12

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

13
14

14 Description

15 [Note: This method overrides **System.IO.TextWriter.Write**.]

16 Exceptions

17
18

Exception	Condition
System.ObjectDisposedException	The writer is closed.

19
20
21

1 StringWriter.Encoding Property

```
2 [ILASM]
3 .property class System.Text.Encoding Encoding { public
4 hidebysig virtual specialname class System.Text.Encoding
5 get_Encoding() }
6
7 [C#]
8 public override 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.StringWriter**. This allows XML code to consume an
19 arbitrary **System.IO.StringWriter** and generate a correct XML
20 header.

21