

# 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 00]
- *Version:* 2.0.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.

# StringWriter() Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor()  
  
[C#]  
public StringWriter()
```

## Summary

Constructs and initializes a new instance of the `System.IO.StringWriter` class.

## Description

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

# StringWriter(System.IFormatProvider)

## Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor(class  
System.IFormatProvider formatProvider)  
  
[C#]  
public StringWriter(IFormatProvider formatProvider)
```

## Summary

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

## Parameters

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

# StringWriter(System.Text.StringBuilder)

## Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Text.StringBuilder sb)

[C#]
public StringWriter(StringBuilder sb)
```

### Summary

Constructs and initializes a new instance of the `System.IO.StringWriter` class that writes to the specified `System.Text.StringBuilder`.

### Parameters

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

### Exceptions

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

# StringWriter(System.Text.StringBuilder, System.IFormatProvider) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Text.StringBuilder sb, class System.IFormatProvider formatProvider)

[C#]
public StringWriter(StringBuilder sb, IFormatProvider formatProvider)
```

## Summary

Constructs and initializes a new instance of the `System.IO.StringWriter` class that writes to the specified `System.Text.StringBuilder` and has the specified format provider.

## Parameters

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

## Exceptions

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

# StringWriter.Close() Method

```
[ILAsm]  
.method public hidebysig virtual void Close()  
  
[C#]  
public override void Close()
```

## Summary

Closes the current `System.IO.StringWriter` instance.

## Description

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

This method overrides `System.IO.Stream.Close`.

]

# StringWriter.Dispose(System.Boolean)

## Method

```
[ILAsm]  
.method family hidebysig virtual void Dispose(bool disposing)  
  
[C#]  
protected override void Dispose(bool disposing)
```

### Summary

Releases the unmanaged resources used by the `System.IO.StringWriter` and optionally releases the managed resources.

### Parameters

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

### Description

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

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

# StringWriter.GetStringBuilder() Method

```
[ILAsm]  
.method public hidebysig virtual class System.Text.StringBuilder  
GetStringBuilder()  
  
[C#]  
public virtual StringBuilder GetStringBuilder()
```

## Summary

Returns the underlying `System.Text.StringBuilder`.

## Return Value

The underlying `System.Text.StringBuilder`.

## Description

This method returns either the `System.Text.StringBuilder` that was passed to the constructor, or the `System.Text.StringBuilder` that was automatically created.

## Behaviors

As described above.



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

## 12 Description

13 [*Note:* This method overrides System.Object.ToString.]  
14  
15  
16

# StringWriter.Write(System.String) Method

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

## Summary

Writes a string to the current instance of the `System.IO.StringWriter`.

## Parameters

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

## Description

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

## Exceptions

Exception	Condition
<code>System.ObjectDisposedException</code>	The writer is closed.

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

```
[ILAsm]  
.method public hidebysig virtual void Write(class System.Char[] buffer,  
int32 index, int32 count)  
  
[C#]  
public override void Write(char[] buffer, int index, int count)
```

## Summary

Writes the specified region of a character array to this instance of the `System.IO.StringWriter`.

## Parameters

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

## Description

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

## Exceptions

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

**System.ObjectDisposedException**

The writer is closed.

1

2

# StringWriter.Write(System.Char) Method

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

## Summary

Writes a character to the current instance of the `System.IO.StringWriter`.

## Parameters

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

## Description

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

## Exceptions

Exception	Condition
<code>System.ObjectDisposedException</code>	The writer is closed.

# StringWriter.Encoding Property

```
[ILAsm]  
.property class System.Text.Encoding Encoding { public hidebysig virtual  
specialname class System.Text.Encoding get_Encoding() }  
  
[C#]  
public override Encoding Encoding { get; }
```

## Summary

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

## Property Value

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

## Behaviors

As described above.

## Usage

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