

# System.IO.StringReader Class

```
[ILAsm]
.class public serializable StringReader extends System.IO.TextReader

[C#]
public class StringReader: TextReader
```

## 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.TextReader` that reads from a string.

## Inherits From: `System.IO.TextReader`

**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.

# StringReader(System.String) Constructor

```
[ILAsm]  
public rtspecialname specialname instance void .ctor(string s)  
  
[C#]  
public StringReader(string s)
```

## Summary

Constructs and initializes a new instance of the `System.IO.StringReader` class that reads from the specified string.

## Parameters

Parameter	Description
<code>s</code>	The <code>System.String</code> to be initialized to.

## Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<code>s</code> is null.

# StringReader.Close() Method

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

## Summary

Closes the System.IO.StringReader.

## Description

Following a call to System.IO.StringReader.Close, other System.IO.StringReader methods on the current instance will throw an exception.

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

This method overrides System.IO.Stream.Close.

]

# StringReader.Dispose(System.Boolean)

## Method

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

### Summary

Releases system resources used by the current instance.

### 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.StringReader` references. This method invokes the `Dispose()` method of each referenced object.

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

# StringReader.Peek() Method

```
[ILAsm]  
.method public hidebysig virtual int32 Peek()  
  
[C#]  
public override int Peek()
```

## Summary

Returns the next available character but does not advance the reader's position in the underlying string.

## Return Value

The next character to be read as a `System.Int32`, or -1 if no more characters are available.

## Description

The current position of the `System.IO.StringReader` is not changed by this operation.

[*Note:* This method returns -1 is when the end of the underlying string is reached because a Unicode character can contain only values between hexadecimal 0x0000 to 0xFFFF (0 to 65535).

This method overrides `System.IO.TextReader.Peek`.

]

## Exceptions

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

# StringReader.Read(System.Char[], System.Int32, System.Int32) Method

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

## Summary

Reads a block of characters from the input string.

## Parameters

Parameter	Description
<i>buffer</i>	A <code>System.Char</code> array. When this method returns, contains the specified character array with the values between <i>index</i> and <i>(index + count - 1)</i> replaced by the characters read from the current source.
<i>index</i>	A <code>System.Int32</code> that specifies the starting index in the buffer.
<i>count</i>	A <code>System.Int32</code> that specifies the number of characters to read.

## Return Value

A `System.Int32` containing the total number of characters read into the buffer, or zero if the end of the underlying string has been reached.

## Description

[*Note:* This method overrides `System.IO.TextReader.Read`.]

## Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>buffer</i> is null.
<code>System.ArgumentException</code>	<i>(index + count) &gt; buffer.Length</i> .

<b>System.ArgumentOutOfRangeException</b>	<i>index</i> < 0  - or-  <i>count</i> < 0.
<b>System.ObjectDisposedException</b>	The current reader is closed.

1

2

# StringReader.Read() Method

```
[ILAsm]  
.method public hidebysig virtual int32 Read()  
  
[C#]  
public override int Read()
```

## Summary

Reads the next character from the input string and advances the character position by one character.

## Return Value

The next character from the underlying string as a `System.Int32`, or -1 if no more characters are available.

## Description

[*Note:* This method returns -1 is when the end of the underlying string is reached because a Unicode character can contain only values between hexadecimal 0x0000 to 0xFFFF (0 to 65535).

This method overrides `System.IO.TextReader.Read`.

]

## Exceptions

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



# StringReader.ReadLine() Method

```
[ILAsm]  
.method public hidebysig virtual string ReadLine()  
  
[C#]  
public override string ReadLine()
```

## Summary

Reads a line from the underlying string.

## Return Value

A `System.String` containing the next line from the underlying string, or `null` if the end of the underlying string is reached.

## Description

A line is defined as a sequence of characters followed by a carriage return (0x000d), a line feed (0x000a), or a carriage return immediately followed by a line feed. The resulting string does not contain the terminating character(s).

[*Note:* This method overrides `System.IO.TextReader.ReadLine`.]

## Exceptions

Exception	Condition
<b>System.ObjectDisposedException</b>	The current reader is closed.
<b>System.OutOfMemoryException</b>	There is insufficient memory to allocate a buffer for the returned string.

# StringReader.ReadToEnd() Method

```
[ILAsm]  
.method public hidebysig virtual string ReadToEnd()  
  
[C#]  
public override string ReadToEnd()
```

## Summary

Returns the underlying string from the current position to the end.

## Return Value

A `System.String` containing the content from the current position to the end of the underlying string.

## Description

[*Note:* This method overrides `System.IO.TextReader.ReadToEnd.`]

## Exceptions

Exception	Condition
<b>System.ObjectDisposedException</b>	The current reader is closed.
<b>System.OutOfMemoryException</b>	There is insufficient memory to allocate a buffer for the returned string.