

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]
- *Version:* 1.0.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
s	The System.String to be initialized to.

Exceptions

Exception	Condition
System.ArgumentNullException	s is null .

1 StringReader.Close() Method

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

6 Summary

7 Closes the **System.IO.StringReader**.

8 Description

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

12
13 [Note: This version of **System.IO.StringReader.Close** is equivalent
14 to **System.IO.StringReader.Dispose(true)**.

15
16 This method overrides **System.IO.Stream.Close**.]

17

1 StringReader.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 system resources used by the current instance.

10 Parameters

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

14 Description

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

19
20 [Note: **System.IO.StringReader.Dispose** may be called multiple
21 times by other objects. When overriding
22 **System.IO.StringReader.Dispose(System.Boolean)**, be careful
23 not to reference objects that have been previously disposed in an
24 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
System.ObjectDisposedException	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 System.Char array. When this method returns, contains the specified character array with the values between <i>index</i> and (<i>index</i> + <i>count</i> - 1) replaced by the characters read from the current source.
<i>index</i>	A System.Int32 that specifies the starting index in the buffer.
<i>count</i>	A System.Int32 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
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentException	(<i>index</i> + <i>count</i>) > <i>buffer</i> .Length.
System.ArgumentOutOfRangeException	<i>index</i> < 0 - or -

1
2
3

	<i>count</i> < 0.
System.ObjectDisposedException	The current reader is closed.

1 StringReader.Read() Method

```
2 [ILASM]
3 .method public hidebysig virtual int32 Read()
4
5 [C#]
6 public override int Read()
```

6 Summary

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

9 Return Value

10
11 The next character from the underlying string as a **System.Int32**, or
12 -1 if no more characters are available.

13 Description

14 [Note: This method returns -1 is when the end of the underlying string
15 is reached because a Unicode character can contain only values
16 between hexadecimal 0x0000 to 0xFFFF (0 to 65535).
17
18 This method overrides **System.IO.TextReader.Read.**]

19 Exceptions

20
21

Exception	Condition
System.ObjectDisposedException	The current reader is closed.

22
23
24

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
System.ObjectDisposedException	The current reader is closed.
System.OutOfMemoryException	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
System.ObjectDisposedException	The current reader is closed.
System.OutOfMemoryException	There is insufficient memory to allocate a buffer for the returned string.