

1 System.IO.StreamReader Class

2
3

```
4 [ILASM]  
5 .class public serializable StreamReader extends  
6 System.IO.TextReader  
  
7 [C#]  
8 public class StreamReader: TextReader
```

9 Assembly Info:

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

15 Implements:

- 16 • System.IDisposable

17 Summary

18

19 Implements a **System.IO.Stream** that reads characters from a byte
20 stream in a particular encoding.

21 Inherits From: System.IO.TextReader

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 The **System.IO.StreamReader** class is designed for character input
30 in a particular **System.Text.Encoding**, whereas subclasses of
31 **System.IO.Stream** are designed for byte input and output.

32

33 [Note: **System.IO.StreamReader** defaults to UTF-8 encoding unless
34 specified otherwise, instead of defaulting to the ANSI code page for
35 the current system. UTF-8 handles Unicode characters correctly and
36 provides consistent results on localized versions of the operating
37 system.

38

39 When reading from a **System.IO.Stream**, it is more efficient to use a
40 buffer that is the same size as the internal buffer of the stream.

41

1 By default, a **System.IO.StreamReader** is not thread safe. For a
2 thread-safe wrapper, see **System.IO.TextReader.Synchronized.**]

3

1 StreamReader(System.IO.Stream)

2 Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.IO.Stream stream)  
  
6 [C#]  
7 public StreamReader(Stream stream)
```

8 Summary

9 Constructs and initializes a new instance of the
10 **System.IO.StreamReader** class for the specified stream.

11 Parameters

12
13

Parameter	Description
<i>stream</i>	The System.IO.Stream to read.

14
15

15 Description

16 This constructor initializes the
17 **System.IO.StreamReader.CurrentEncoding** property to
18 **System.Text.UTF8Encoding**, the
19 **System.IO.StreamReader.BaseStream** property using *stream*, and
20 the internal buffer to the default size. [*Note:* The default buffer size is
21 implementation dependent.]

22 Exceptions

23
24

Exception	Condition
System.ArgumentException	<i>stream</i> does not support reading.
System.ArgumentNullException	<i>stream</i> is null .

25
26
27

1 StreamReader(System.IO.Stream, 2 System.Boolean) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.IO.Stream stream, bool  
6 detectEncodingFromByteOrderMarks)  
  
7 [C#]  
8 public StreamReader(Stream stream, bool  
9 detectEncodingFromByteOrderMarks)
```

10 Summary

11 Constructs and initializes a new instance of the
12 **System.IO.StreamReader** class for the specified stream, with the
13 specified byte order mark detection option.

14 Parameters

15
16

Parameter	Description
<i>stream</i>	The <i>stream</i> to read.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .

17

18 Description

19 This constructor initializes the
20 **System.IO.StreamReader.CurrentEncoding** property to
21 **System.Text.UTF8Encoding**, the
22 **System.IO.StreamReader.BaseStream** property using *stream*, and
23 the internal buffer to the default size. [*Note*: The default buffer size is
24 implementation defined.]

25

26 If requested, the current constructor detects the encoding by
27 examining the first three bytes of the stream. The constructor
28 automatically recognizes UTF-8, little-endian Unicode, and big-endian
29 Unicode text if the file starts with the appropriate byte order marks.
30 Otherwise, UTF-8 encoding is used. For more information, see the
31 **System.Text.Encoding.GetPreamble** method.

1 **Exceptions**

2

3

Exception	Condition
System.ArgumentException	<i>stream</i> does not support reading.
System.ArgumentNullException	<i>stream</i> is null .

4

5

6

1 StreamReader(System.IO.Stream, 2 System.Text.Encoding) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.IO.Stream stream, class System.Text.Encoding  
6 encoding)  
  
7 [C#]  
8 public StreamReader(Stream stream, Encoding encoding)
```

9 Summary

10 Constructs and initializes a new instance of the
11 **System.IO.StreamReader** class for the specified stream with the
12 specified character encoding.

13 Parameters

14
15

Parameter	Description
<i>stream</i>	The System.IO.Stream to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.

16
17

Description

18 This constructor initializes the
19 **System.IO.StreamReader.CurrentEncoding** property using
20 *encoding*, the **System.IO.StreamReader.BaseStream** property
21 using *stream*, and the internal buffer to the default size. [Note: The
22 default buffer size is implementation defined.]

23 Exceptions

24
25

Exception	Condition
System.ArgumentException	<i>stream</i> does not support reading.
System.ArgumentNullException	<i>stream</i> or <i>encoding</i> is null .

26
27
28

1 StreamReader(System.IO.Stream, 2 System.Text.Encoding, System.Boolean) 3 Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(class  
6 System.IO.Stream stream, class System.Text.Encoding  
7 encoding, bool detectEncodingFromByteOrderMarks)
```

```
8 [C#]  
9 public StreamReader(Stream stream, Encoding encoding, bool  
10 detectEncodingFromByteOrderMarks)
```

11 Summary

12 Constructs and initializes a new instance of the
13 **System.IO.StreamReader** class for the specified stream, with the
14 specified character encoding and byte order mark detection option.

15 Parameters

Parameter	Description
<i>stream</i>	The System.IO.Stream to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .

19 Description

20 This constructor initializes the
21 **System.IO.StreamReader.CurrentEncoding** property using
22 *encoding*, the **System.IO.StreamReader.BaseStream** property
23 using *stream*, and the internal buffer to the default size. [Note: The
24 default buffer size is implementation defined.]

25
26 If requested, this constructor detects the encoding by examining the
27 first three bytes of *stream*. This constructor automatically recognizes
28 UTF-8, little-endian Unicode, and big-endian Unicode text if the stream
29 starts with the appropriate byte order marks. Otherwise, the user-
30 provided encoding is used. See the
31 **System.Text.Encoding.GetPreamble** method for more information.

1 **Exceptions**

2

3

Exception	Condition
System.ArgumentException	<i>stream</i> does not support reading.
System.ArgumentNullException	<i>stream</i> or <i>encoding</i> is null .

4

5

6

1 StreamReader(System.IO.Stream, 2 System.Text.Encoding, System.Boolean, 3 System.Int32) Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(class  
6 System.IO.Stream stream, class System.Text.Encoding  
7 encoding, bool detectEncodingFromByteOrderMarks, int32  
8 bufferSize)  
9  
10 [C#]  
11 public StreamReader(Stream stream, Encoding encoding, bool  
12 detectEncodingFromByteOrderMarks, int bufferSize)
```

12 Summary

13 Constructs and initializes a new instance of the
14 **System.IO.StreamReader** class for the specified stream, with the
15 specified character encoding, byte order mark detection option, and
16 buffer size.

17 Parameters

18
19

Parameter	Description
<i>stream</i>	The System.IO.Stream to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .
<i>bufferSize</i>	A System.Int32 that specifies the minimum buffer size, in number of 16-bit characters. If <i>bufferSize</i> is less than the minimum allowable size (128 characters), the minimum allowable size is used.

20
21

21 Description

22 This constructor initializes the
23 **System.IO.StreamReader.CurrentEncoding** property using
24 *encoding* parameter the **System.IO.StreamReader.BaseStream**
25 property using *stream*.
26

1 If requested, this constructor detects the encoding by examining the
2 first three bytes of the stream. The constructor automatically
3 recognizes UTF-8, little-endian Unicode, and big-endian Unicode text if
4 the file starts with the appropriate byte order marks. Otherwise, the
5 user-provided encoding is used. For more information, see the
6 **System.Text.Encoding.GetPreamble** method.

7
8 [Note: When reading from a **System.IO.Stream**, it is more efficient
9 to use a buffer that is the same size as the internal buffer of the
10 stream.]

11 **Exceptions**

12
13

Exception	Condition
System.ArgumentException	<i>stream</i> does not support reading.
System.ArgumentNullException	<i>stream</i> or <i>encoding</i> is null .
System.ArgumentOutOfRangeException	<i>bufferSize</i> is less than or equal to zero.

14
15
16

1 StreamReader(System.String) Constructor

```
2 [ILASM]
3 public rtspecialname specialname instance void .ctor(string
4 path)
5
6 [C#]
7 public StreamReader(string path)
```

7 Summary

8 Constructs and initializes a new instance of the
9 **System.IO.StreamReader** class for the specified file name.

10 Parameters

11
12

Parameter	Description
<i>path</i>	A System.String that specifies the complete file path to read.

13
14

14 Description

15 This constructor initializes the
16 **System.IO.StreamReader.CurrentEncoding** property to
17 **System.Text.UTF8Encoding**, and the internal buffer to the default
18 size. [Note: The default buffer size is implementation defined.]

19
20 [Note: *path* is not required to be a file stored on disk; it can be any
21 part of a system that supports access via streams. For example,
22 depending on the system, this class may be able to access a physical
23 device.

24
25 For information on the valid format and characters for path strings,
26 see **System.IO.Path**.]

27 Exceptions

28
29

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.IO.FileNotFoundException	The file specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> is null .

1
2
3

1 StreamReader(System.String, 2 System.Boolean) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(string  
5 path, bool detectEncodingFromByteOrderMarks)  
  
6 [C#]  
7 public StreamReader(string path, bool  
8 detectEncodingFromByteOrderMarks)
```

9 Summary

10 Constructs and initializes a new instance of the
11 **System.IO.StreamReader** class for the specified file name, with the
12 specified byte order mark detection option.

13 Parameters

14
15

Parameter	Description
<i>path</i>	A System.String that specifies the complete file path to read.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .

16

17 Description

18 This constructor initializes the
19 **System.IO.StreamReader.CurrentEncoding** property to
20 **System.Text.UTF8Encoding**, and the internal buffer to the default
21 size. [Note: The default buffer size is implementation defined.]

22

23 If requested, the current constructor detects the encoding by
24 examining the first three bytes of the stream. The constructor
25 automatically recognizes UTF-8, little-endian Unicode, and big-endian
26 Unicode text if the file starts with the appropriate byte order marks.
27 Otherwise, UTF-8 encoding is used. See the
28 **System.Text.Encoding.GetPreamble** method for more information.

29

30 [Note: *path* is not required to be a file stored on disk; it can be any
31 part of a system that supports access via streams. For example,
32 depending on the system, this class may be able to access a physical
33 device.

1
2
3

For information on the valid format and characters for path strings,
see **System.IO.Path.**]

4 **Exceptions**

5
6

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.IO.FileNotFoundException	The file specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> is null .

7
8
9

1 StreamReader(System.String, 2 System.Text.Encoding) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(string  
5 path, class System.Text.Encoding encoding)  
  
6 [C#]  
7 public StreamReader(string path, Encoding encoding)
```

8 Summary

9 Constructs and initializes a new instance of the
10 **System.IO.StreamReader** class for the specified file name and with
11 the specified character encoding.

12 Parameters

13
14

Parameter	Description
<i>path</i>	A System.String that specifies the complete file path to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.

15
16

16 Description

17 This constructor initializes the
18 **System.IO.StreamReader.CurrentEncoding** property using
19 *encoding*, and the internal buffer to the default size. [*Note*: The default
20 buffer size is implementation defined.]

21
22 [*Note*: *path* is not required to be a file stored on disk; it can be any
23 part of a system that supports access via streams. For example,
24 depending on the system, this class may be able to access a physical
25 device.

26
27 For information on the valid format and characters for path strings,
28 see **System.IO.Path**.]

29 Exceptions

30
31

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.

System.IO.FileNotFoundException	The file specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> or <i>encoding</i> is null .

1
2
3

StreamReader(System.String, System.Text.Encoding, System.Boolean) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(string
path, class System.Text.Encoding encoding, bool
detectEncodingFromByteOrderMarks)
```

```
[C#]
public StreamReader(string path, Encoding encoding, bool
detectEncodingFromByteOrderMarks)
```

Summary

Constructs and initializes a new instance of the **System.IO.StreamReader** class for the specified file name, with the specified character encoding and byte order mark detection option.

Parameters

Parameter	Description
<i>path</i>	A System.String that specifies the complete file path to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .

Description

This constructor initializes the **System.IO.StreamReader.CurrentEncoding** property using *encoding*, and the internal buffer to the default size. [Note: The default buffer size is implementation defined.]

If requested, the current constructor detects the encoding by examining the first three bytes of the stream. The constructor automatically recognizes UTF-8, little-endian Unicode, and big-endian Unicode text if the file starts with the appropriate byte order marks. Otherwise, the user-provided encoding is used. See the **System.Text.Encoding.GetPreamble** method for more information.

1
2
3
4
5
6
7
8
9
10
11

[*Note*: *path* is not required to be a file stored on disk; it can be any part of a system that supports access via streams. For example, depending on the system, this class may be able to access a physical device.

For information on the valid format and characters for path strings, see **System.IO.Path**.]

Exceptions

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.IO.FileNotFoundException	The file specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> or <i>encoding</i> is null .

12
13
14

1 StreamReader(System.String, 2 System.Text.Encoding, System.Boolean, 3 System.Int32) Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(string  
6 path, class System.Text.Encoding encoding, bool  
7 detectEncodingFromByteOrderMarks, int32 bufferSize)
```

```
8 [C#]  
9 public StreamReader(string path, Encoding encoding, bool  
10 detectEncodingFromByteOrderMarks, int bufferSize)
```

11 Summary

12 Constructs and initializes a new instance of the
13 **System.IO.StreamReader** class for the specified file name, with the
14 specified character encoding, byte order mark detection option, and
15 buffer size.

16 Parameters

17
18

Parameter	Description
<i>path</i>	A System.String that specifies the complete file path to read.
<i>encoding</i>	A System.Text.Encoding that specifies the character encoding to use.
<i>detectEncodingFromByteOrderMarks</i>	A System.Boolean value that indicates whether the new System.IO.StreamReader is required to look for byte order marks at the beginning of the stream. Specify true to enable detection of byte order marks; otherwise, specify false .
<i>bufferSize</i>	A System.Int32 that specifies the minimum buffer size, in number of 16-bit characters. If less than the minimum allowable size (128 characters), the minimum allowable size is used.

19

20 Description

21 This constructor initializes the
22 **System.IO.StreamReader.CurrentEncoding** property using
23 *encoding*.

24
25 If requested, the current constructor detects the encoding by

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

examining the first three bytes of the stream. The constructor automatically recognizes UTF-8, little-endian Unicode, and big-endian Unicode text if the file starts with the appropriate byte order marks. Otherwise, the user-provided encoding is used. See the **System.Text.Encoding.GetPreamble** method for more information.

[*Note:* *path* is not required to be a file stored on disk; it can be any part of a system that supports access via streams. For example, depending on the system, this class may be able to access a physical device.

When reading from a **System.IO.Stream**, it is more efficient to use a buffer that is the same size as the internal buffer of the stream.

For information on the valid format and characters for path strings, see **System.IO.Path**.]

17 **Exceptions**
18
19

Exception	Condition
System.IO.IOException	<i>path</i> is in an invalid format or contains invalid characters.
System.IO.DirectoryNotFoundException	The directory information specified in <i>path</i> was not found.
System.IO.FileNotFoundException	The file specified in <i>path</i> was not found.
System.ArgumentException	<i>path</i> is an empty string ("").
System.ArgumentNullException	<i>path</i> or <i>encoding</i> is null .
System.ArgumentOutOfRangeException	<i>bufferSize</i> is less than or equal to zero.

20
21
22

1 StreamReader.Close() Method

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

6 Summary

7 Closes the current instance of **System.IO.StreamReader**, releasing
8 any system resources associated with it.

9 Description

10 Following a call to this method, operations on the current instance
11 might raise exceptions.

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

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

17

1 StreamReader.DiscardBufferedData() 2 Method

```
3 [ILASM]  
4 .method public hidebysig instance void  
5 DiscardBufferedData()
```

```
6 [C#]  
7 public void DiscardBufferedData()
```

8 Summary

9 Allows a **System.IO.StreamReader** to discard its buffered data.

10 Description

11 [*Note:* This method is useful when reading from a stream after seeking
12 to a new position. If this method is not called and the internal buffer is
13 not empty, a read attempt at the new location will first return data
14 that is in the buffer before returning the text at the current position in
15 the stream.]

16

1 StreamReader.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.StreamReader** 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

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.StreamReader** references. This method invokes the
20 **Dispose()** method of each referenced object.
21

22 [Note: **System.IO.StreamReader.Dispose** may be called multiple
23 times by other objects. When overriding
24 **System.IO.StreamReader.Dispose(System.Boolean)**, be careful
25 not to reference objects that have been previously disposed in an
26 earlier call to **System.IO.StreamReader.Dispose**.
27

28 This method calls the dispose method of the base class,
29 **System.IO.TextReader.Dispose(*disposing*)**.]
30

1 StreamReader.Peek() Method

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

6 Summary

7 Returns the next character in the underlying stream without advancing
8 the position of the **System.IO.StreamReader** in the stream.

9 Return Value

10

11 The next character from the character source as a **System.Int32**, or -
12 1 if at the end of the stream.

13 Description

14 [*Note:* This method returns -1 is when the end of the underlying
15 stream 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.Peek.**]

19 Exceptions

20

21

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

22

23

24

1 StreamReader.Read(System.Char[], 2 System.Int32, System.Int32) Method

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

9 Summary

10 Reads a maximum of *count* characters from the current stream into
11 *buffer*, beginning at *index*.

12 Parameters

13
14

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 instance.
<i>index</i>	A System.Int32 that specifies the index of <i>buffer</i> at which to begin writing.
<i>count</i>	A System.Int32 that specifies the maximum number of characters to read.

15
16
17

Return Value

18 A **System.Int32** containing the number of characters that have been
19 read, or zero if there are no more characters left to read. Can be less
20 than *count* if the end of the stream has been reached.

21 Description

22 [Note: This method returns after either *count* characters are read, or
23 the end of the file is reached. **System.IO.TextReader.ReadBlock** is
24 a blocking version of **System.IO.StreamReader.Read**.

25
26

This method overrides **System.IO.TextReader.Read**.]

27 Exceptions

28
29

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

System.ArgumentException	<i>buffer.Length - index < count.</i>
System.ArgumentNullException	<i>buffer is null.</i>
System.ArgumentOutOfRangeException	<i>index or count is negative.</i>
System.IO.IOException	An I/O error occurred -or- the current stream is closed.

1
2
3

1 StreamReader.Read() Method

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

6 Summary

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

9 Return Value

10

11 The next character from the character source represented as a
12 **System.Int32**, or -1 if at the end of the stream.

13 Description

14 [*Note:* This method returns -1 is when the end of the underlying
15 stream 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.IO.IOException	An I/O error occurred.

22

23

24

1 StreamReader.ReadLine() Method

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

6 Summary

7 Reads a line of characters from the current stream and returns the
8 data as a string.

9 Return Value

10

11 A **System.String** containing the next line from the input stream, or
12 **null** if the end of the input stream is reached.

13 Description

14 [Note: This method defines a line as a sequence of characters followed
15 by a carriage return (hexadecimal 0x000d), a line feed (hexadecimal
16 0x000a), or **System.Environment.NewLine**. The returned string
17 does not contain the terminating character(s).
18

19 This method overrides **System.IO.TextReader.ReadLine.**]

20 Exceptions

21

22

Exception	Condition
System.IO.IOException	An I/O error occurred.
System.OutOfMemoryException	There is insufficient memory to allocate a buffer for the returned string.

23

24

25

1 StreamReader.ReadToEnd() Method

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

6 Summary

7 Reads the stream from the current position to the end of the stream.

8 Return Value

9

10 A **System.Int32** containing the rest of the stream as a string, from
11 the current position to the end. If the current position is at the end of
12 the stream, returns the empty string ("").

13 Description

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

15 Exceptions

16

17

Exception	Condition
System.IO.IOException	An I/O error occurred.
System.OutOfMemoryException	There is insufficient memory to allocate a buffer for the returned string.

18

19

20

1 StreamReader.BaseStream Property

```
2 [ILASM]
3 .property class System.IO.Stream BaseStream { public
4 hidebysig virtual specialname class System.IO.Stream
5 get_BaseStream() }
6
7 [C#]
8 public virtual Stream BaseStream { get; }
```

8 Summary

9 Gets the underlying stream.

10 Property Value

11

12 The underlying **System.IO.Stream** which the current
13 **System.IO.StreamReader** instance is reading.

14 Description

15 This property is read-only.

16

1 StreamReader.CurrentEncoding Property

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

8 Summary

9 Gets the current character encoding that the current
10 **System.IO.StreamReader** is using.

11 Property Value

12

13 The current **System.Text.Encoding** used by the current reader.

14 Description

15 This property is read-only.

16

17 The value returned by this property might change after the first call to
18 a **System.IO.StreamReader.Read** method if encoding auto detection
19 was specified to the constructor for the current instance.

20