

System.IO.MemoryStream Class

```
[ILASM]
.class public serializable MemoryStream extends
System.IO.Stream

[C#]
public class MemoryStream: Stream
```

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

Provides support for creating and using a stream whose backing store is memory.

Inherits From: System.IO.Stream

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.

Description

The **System.IO.MemoryStream** class creates streams that have memory as a backing store instead of a disk or a network connection. **System.IO.MemoryStream** encapsulates data stored as an unsigned byte array. The encapsulated data is directly accessible in memory. Memory streams can reduce the need for temporary buffers and files in an application.

The *current position* of a stream is the position at which the next read or write operation takes place. The current position can be retrieved or set through the **System.IO.MemoryStream.Seek** method. When a new instance of **System.IO.MemoryStream** is created, the current position is set to zero.

1 [Note: Memory streams created with an unsigned byte array provide a
2 non-resizable stream view of the data. When using a byte array, you
3 can neither append to nor shrink the stream, although you might be
4 able to modify the existing contents depending on the parameters
5 passed into the constructor.]

6

1 MemoryStream() Constructor

```
2 [ILASM]  
3 public rtSpecialName specialName instance void .ctor()  
4 [C#]  
5 public MemoryStream()
```

6 Summary

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

9 Description

10 The **System.IO.Stream.CanRead**, **System.IO.Stream.CanSeek**,
11 and **System.IO.Stream.CanWrite** properties of the new instance of
12 the **System.IO.MemoryStream** class are set to **true**.

13
14 The capacity of the new stream instance can be increased by using the
15 **System.IO.MemoryStream.SetLength** method or by setting the
16 **System.IO.MemoryStream.Capacity** property.

17
18 The new stream exposes the underlying byte buffer, which can be
19 accessed through the **System.IO.MemoryStream.GetBuffer**
20 method.

21

1 MemoryStream(System.Int32)

2 Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(int32  
5 capacity)  
  
6 [C#]  
7 public MemoryStream(int capacity)
```

8 Summary

9 Constructs and initializes a new resizable instance of the
10 **System.IO.MemoryStream** class.

11 Parameters

12
13

Parameter	Description
<i>capacity</i>	A System.Int32 that specifies the initial size of the internal System.Byte array.

14
15

15 Description

16 The **System.IO.Stream.CanRead**, **System.IO.Stream.CanSeek**,
17 and **System.IO.Stream.CanWrite** properties of the new instance of
18 the **System.IO.MemoryStream** class are set to **true**.

19
20 The **System.IO.MemoryStream.Capacity** of the new stream
21 instance is set to *capacity* can be increased by using the
22 **System.IO.MemoryStream.SetLength** method or by setting the
23 **System.IO.MemoryStream.Capacity** property. Write operations at
24 the end of the new instance of the **System.IO.MemoryStream** class
25 expand the **System.IO.MemoryStream**.

26
27 The new stream exposes the underlying byte buffer, which can be
28 accessed through the **System.IO.MemoryStream.GetBuffer**
29 method.

30 Exceptions

31
32

Exception	Condition
System.ArgumentOutOfRangeException	<i>capacity</i> is negative.

33
34
35

1 MemoryStream(System.Byte[])

2 Constructor

```
3 [ILASM]
4 public rtspecialname specialname instance void .ctor(class
5 System.Byte[] buffer)
6
7 [C#]
8 public MemoryStream(byte[] buffer)
```

8 Summary

9 Constructs and initializes a new non-resizable instance of the
10 **System.IO.MemoryStream** class.

11 Parameters

12
13

Parameter	Description
<i>buffer</i>	The System.Byte array from which to create the new stream.

14

15 Description

16 The **System.IO.Stream.CanRead**, **System.IO.Stream.CanSeek**,
17 and **System.IO.Stream.CanWrite** properties of the new instance of
18 the **System.IO.MemoryStream** class are set to **true**.
19 **System.IO.MemoryStream.Capacity** is set to the length of the
20 specified **System.Byte** array.

21

22 [*Note:* The new stream instance can be written to, but the
23 **System.IO.MemoryStream.Capacity** of the underlying
24 **System.Byte** array cannot be changed. The length of the stream
25 cannot be set to a value greater than
26 **System.IO.MemoryStream.Capacity**, but the stream can be
27 truncated (see **System.IO.MemoryStream.SetLength**).]

28

29 The new stream does not expose the underlying byte buffer, and calls
30 to the **System.IO.MemoryStream.GetBuffer** method throw
31 **System.UnauthorizedAccessException**.

32 Exceptions

33
34

Exception	Condition
System.ArgumentNullException	The <i>buffer</i> parameter is null .

35

36

37

1 MemoryStream(System.Byte[], 2 System.Boolean) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.Byte[] buffer, bool writable)  
  
6 [C#]  
7 public MemoryStream(byte[] buffer, bool writable)
```

8 Summary

9 Constructs and initializes a new non-resizable instance of the
10 **System.IO.MemoryStream** class.

11 Parameters

12
13

Parameter	Description
<i>buffer</i>	The System.Byte array from which to create the new stream.
<i>writable</i>	A System.Boolean that specifies whether the new stream instance supports writing.

14
15

Description

16 The **System.IO.Stream.CanRead** and **System.IO.Stream.CanSeek**
17 properties of the new instance of the **System.IO.MemoryStream**
18 class are set to **true**. The **System.IO.MemoryStream.Capacity**
19 property is set to the length of the specified **System.Byte** array. The
20 **System.IO.Stream.CanWrite** property is set to *writable*.

21
22
23
24
25
26
27
28
29
30
31

[Note: The new stream instance can be written to, but the
System.IO.MemoryStream.Capacity of the underlying
System.Byte array cannot be changed. The length of the stream
cannot be set to a value larger than
System.IO.MemoryStream.Capacity, but the stream can be
truncated (see **System.IO.MemoryStream.SetLength**).]

The new stream does not expose the underlying **System.Byte** buffer,
and calls to the **System.IO.MemoryStream.GetBuffer** method
throw **System.UnauthorizedAccessException**.

32 Exceptions

33
34

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

1
2
3

1 MemoryStream(System.Byte[], 2 System.Int32, System.Int32) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(class  
5 System.Byte[] buffer, int32 index, int32 count)  
  
6 [C#]  
7 public MemoryStream(byte[] buffer, int index, int count)
```

8 Summary

9 Constructs and initializes a new non-resizable instance of the
10 **System.IO.MemoryStream** class.

11 Parameters

12
13

Parameter	Description
<i>buffer</i>	The System.Byte array from which to create the new stream.
<i>index</i>	A System.Int32 that specifies the index into <i>buffer</i> at which the stream begins.
<i>count</i>	A System.Int32 that specifies the length of the stream in bytes.

14
15

Description

16 The **System.IO.Stream.CanRead**, **System.IO.Stream.CanSeek**,
17 and **System.IO.Stream.CanWrite** properties of the new
18 **System.IO.MemoryStream** instance are set to **true**. The
19 **System.IO.MemoryStream.Capacity** property is set to *count*.

20

21 [Note: The new stream instance can be written to, but the
22 **System.IO.MemoryStream.Capacity** of the underlying
23 **System.Byte** array cannot be changed. The length of the stream
24 cannot be set to a value larger than
25 **System.IO.MemoryStream.Capacity**, but the stream can be
26 truncated (see **System.IO.MemoryStream.SetLength**).]

27

28 The new stream does not expose the underlying **System.Byte** buffer,
29 and calls to the **System.IO.MemoryStream.GetBuffer** method
30 throw **System.UnauthorizedAccessException**.

31 Exceptions

32
33

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

1
2
3

System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is less than zero.
System.ArgumentException	(<i>index</i> + <i>count</i>) is greater than the length of <i>buffer</i> .

1 MemoryStream(System.Byte[], 2 System.Int32, System.Int32, 3 System.Boolean) Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(class  
6 System.Byte[] buffer, int32 index, int32 count, bool  
7 writable)  
  
8 [C#]  
9 public MemoryStream(byte[] buffer, int index, int count,  
10 bool writable)
```

11 Summary

12 Constructs and initializes a new non-resizable instance of the
13 **System.IO.MemoryStream** class.

14 Parameters

15
16

Parameter	Description
<i>buffer</i>	The System.Byte array from which to create the new stream.
<i>index</i>	A System.Int32 that specifies the index in <i>buffer</i> at which the stream begins.
<i>count</i>	A System.Int32 that specifies the length of the stream in bytes.
<i>writable</i>	A System.Boolean that specifies whether the new stream instance supports writing.

17

18 Description

19 The **System.IO.MemoryStream.CanRead** and
20 **System.IO.MemoryStream.CanSeek** properties of the new
21 **System.IO.MemoryStream** are set to **true**. The
22 **System.IO.MemoryStream.Capacity** property is set to *count*. The
23 **System.IO.Stream.CanWrite** property is set to *writable*.

24

25 [Note: The new stream instance can be written to, but the
26 **System.IO.MemoryStream.Capacity** of the underlying byte array
27 cannot be changed. The length of the stream cannot be set to a value
28 larger than **System.IO.MemoryStream.Capacity**, but the stream
29 can be truncated (see **System.IO.MemoryStream.SetLength**).]

30

31 The new stream does not expose the underlying byte buffer, and calls
32 to the **System.IO.MemoryStream.GetBuffer** method throw
33 **System.UnauthorizedAccessException**.

1 **Exceptions**
2
3

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> are negative.
System.ArgumentException	(<i>index</i> + <i>count</i>) is greater than the length of <i>buffer</i> .

4
5
6

MemoryStream(System.Byte[], System.Int32, System.Int32, System.Boolean, System.Boolean) Constructor

```
[ILASM]  
public rtspecialname specialname instance void .ctor(class  
System.Byte[] buffer, int32 index, int32 count, bool  
writable, bool publiclyVisible)
```

```
[C#]  
public MemoryStream(byte[] buffer, int index, int count,  
bool writable, bool publiclyVisible)
```

Summary

Constructs and initializes a new instance of the **System.IO.MemoryStream** class.

Parameters

Parameter	Description
<i>buffer</i>	The System.Byte array from which to create the new stream.
<i>index</i>	A System.Int32 that specifies the index into <i>buffer</i> at which the stream begins.
<i>count</i>	A System.Int32 that specifies the length of the stream in bytes.
<i>writable</i>	A System.Boolean that specifies whether the new stream instance supports writing.
<i>publiclyVisible</i>	A System.Boolean that specifies whether <i>buffer</i> is exposed via the System.IO.MemoryStream.GetBuffer , which returns the System.Byte array from which the stream was created. Specify true to expose <i>buffer</i> ; otherwise, specify false .

Description

The **System.IO.MemoryStream.CanRead** and **System.IO.MemoryStream.CanSeek** properties of the new **System.IO.MemoryStream** instance are set to **true**. The **System.IO.MemoryStream.Capacity** property is set to *count*. The **System.IO.Stream.CanWrite** property is set to *writable*.

[Note: The new stream instance can be written to, but the **System.IO.MemoryStream.Capacity** of the underlying **System.Byte** array cannot be changed. The length of the stream cannot be set to a value larger than

1 **System.IO.MemoryStream.Capacity**, but the stream can be
2 truncated (see **System.IO.MemoryStream.SetLength**).]

3 **Exceptions**

4
5

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentOutOfRangeException	<i>index</i> or <i>count</i> is negative.
System.ArgumentException	(<i>index</i> + <i>count</i>) is greater than the length of <i>buffer</i> .

6
7
8

1 **MemoryStream.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.MemoryStream** instance.

8 **Description**

9 The stream will not support reading or writing after this method is
10 invoked. Following a call to **System.IO.MemoryStream.Close**,
11 operations on the stream can raise an exception.

12
13 The buffer of a closed **System.IO.MemoryStream** is still available,
14 and the **System.IO.MemoryStream.ToArray** and
15 **System.IO.MemoryStream.GetBuffer** methods can be called
16 successfully.

17
18 [*Note:* This method overrides **System.IO.Stream.Close**.]

19

1 MemoryStream.Flush() Method

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

6 Summary

7 Overrides **System.IO.Stream.Flush** so that no action is performed.

8 Description

9 Since any data written to a **System.IO.MemoryStream** is written
10 into RAM, this method is redundant.

11
12 [*Note:* This method overrides **System.IO.Stream.Flush.**]

13

1 MemoryStream.GetBuffer() Method

```
2 [ILASM]  
3 .method public hidebysig virtual class System.Byte[]  
4 GetBuffer()  
  
5 [C#]  
6 public virtual byte[] GetBuffer()
```

7 Summary

8 Returns the array of unsigned bytes from which this stream was
9 created.

10 Return Value

11

12 The **System.Byte** array from which the current stream was created,
13 or the underlying array if a **System.Byte** array was not provided to
14 the **System.IO.MemoryStream** constructor during construction of
15 the current instance.

16 Description

17 To create a **System.IO.MemoryStream** instance with a publicly
18 visible buffer use the default constructor, or
19 **System.IO.MemoryStream(System.Byte [], System.Int32,**
20 **System.Int32, System.Boolean, System.Boolean)** or
21 **System.IO.MemoryStream(System.Int32)** constructor.
22

23 If the current stream is resizable, multiple calls to this method do not
24 return the same array if the underlying **System.Byte** array is resized
25 between calls. For additional information, see
26 **System.IO.MemoryStream.Capacity**.
27

28 [*Note:* This method works when the **System.IO.MemoryStream** is
29 closed.]

30 Behaviors

31 As described above.

32 Exceptions

33

34

Exception	Condition
System.UnauthorizedAccessException	The current instance was not created with a publicly visible buffer.

1
2 **Example**
3

4 The following example demonstrates that two calls to the
5 **System.IO.MemoryStream.GetBuffer** method on a resizable stream
6 do not return the same array if the underlying byte array is
7 reallocated.

```
8 [C#  
9  
10 using System;  
11 using System.IO;  
12  
13 public class MemoryStreamTest {  
14     public static void Main() {  
15  
16         MemoryStream ms = new MemoryStream(10);  
17  
18         byte[] a = ms.GetBuffer();  
19         byte[] b = ms.GetBuffer();  
20  
21         //Force reallocation of the underlying byte array.  
22         ms.Capacity = 10240;  
23         byte[] c = ms.GetBuffer();  
24  
25  
26         if(Object.ReferenceEquals(a, b))  
27             Console.WriteLine("a and b represent the same  
28 instance.");  
29         else  
30             Console.WriteLine("a and b represent the different  
31 instances.");  
32  
33         if(Object.ReferenceEquals(a, c))  
34             Console.WriteLine("a and c represent the same  
35 instance.");  
36         else  
37             Console.WriteLine("a and c represent the different  
38 instances.");  
39  
40     }  
41 }
```

42 The output is
43
44 a and b represent the same instance.
45
46

1 a and c represent the different instances.
2
3

1 `MemoryStream.Read(System.Byte[],` 2 `System.Int32, System.Int32) Method`

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

9 **Summary**

10 Reads a block of bytes from the current stream at the current position,
11 and writes the data to the specified byte array.

12 **Parameters**

13
14

Parameter	Description
<i>buffer</i>	A System.Byte array. When this method returns, contains the specified byte array with the values between <i>offset</i> and (<i>offset</i> + <i>count</i> - 1) replaced by the characters read from the current stream.
<i>offset</i>	A System.Int32 that specifies the byte offset in <i>buffer</i> at which to begin writing.
<i>count</i>	A System.Int32 that specifies the maximum number of bytes to read.

15
16
17

16 **Return Value**

18 A **System.Int32** that specifies the total number of bytes read into the
19 buffer, or zero if the end of the stream is reached before any bytes are
20 read.

21 **Description**

22 If the read operation is successful, the current position within the
23 stream advances by the number of bytes read. If an exception occurs,
24 the current position within the stream remains unchanged.

25
26
27
28
29
30

[Note: If the byte array specified in the *buffer* parameter is the underlying buffer returned by the **System.IO.MemoryStream.GetBuffer** method, the array contents are overwritten, and no exception is thrown.

31 This method overrides **System.IO.Stream.Read.**]

1 **Exceptions**

2

3

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.ArgumentOutOfRangeException	<i>offset</i> or <i>count</i> is negative.
System.ArgumentException	(<i>offset</i> + <i>count</i>) is larger than the length of <i>buffer</i> .
System.ObjectDisposedException	The current stream is closed.

4

5 **Example**

6

7 The following example demonstrates the result of reading from a
8 **System.IO.MemoryStream** into its underlying byte array.

9

10

[C#]

11

```
using System;
```

12

```
using System.IO;
```

13

14

```
public class MemoryStreamTest {
```

15

```
    public static void Main() {
```

16

```
        byte[] values = new byte [] {0,1,2,3,4,5,6,7,8,9};
```

17

18

```
        foreach (byte b in values) {
```

19

```
            Console.Write(b);
```

20

```
        }
```

21

22

```
        Console.WriteLine();
```

23

24

```
        MemoryStream ms = new MemoryStream (values);
```

25

26

```
        ms.Read(values, 1, 5);
```

27

28

```
        foreach (byte b in values) {
```

29

```
            Console.Write(b);
```

30

```
        }
```

31

```
    }
```

32

```
}
```

33

34

The output is

35

0123456789

36

37

38

1 0012346789
2
3

1 MemoryStream.ReadByte() Method

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

6 Summary

7 Reads a byte from the current stream at the current position.

8 Return Value

9

10 The byte cast to a **System.Int32**, or -1 if the end of the stream has
11 been reached.

12 Description

13 If the read operation is successful, the current position within the
14 stream is advanced by one byte. If an exception occurs, the current
15 position within the stream is unchanged.

16

17 [*Note:* This method overrides **System.IO.Stream.ReadByte.**]

18 Exceptions

19

20

Exception	Condition
System.ObjectDisposedException	The current stream is closed.

21

22

23

1 MemoryStream.Seek(System.Int64, 2 System.IO.SeekOrigin) Method

```
3 [ILASM]  
4 .method public hidebysig virtual int64 Seek(int64 offset,  
5 valuetype System.IO.SeekOrigin loc)  
  
6 [C#]  
7 public override long Seek(long offset, SeekOrigin loc)
```

8 Summary

9 Sets the current position within the current stream to the specified
10 value.

11 Parameters

12
13

Parameter	Description
<i>offset</i>	A System.Int64 that specifies the new position within the stream. This is relative to the <i>loc</i> parameter, and may be positive or negative.
<i>loc</i>	A System.IO.SeekOrigin value that specifies the seek reference point.

14
15
16

Return Value

17 A **System.Int64** containing the new position within the stream,
18 calculated by combining the seek reference point and the offset.

19 Description

20 The position cannot be set to more than one byte beyond the end of
21 the stream.

22
23

[Note: This method overrides **System.IO.Stream.Seek**.]

24 Exceptions

25
26

Exception	Condition
System.IO.IOException	Seeking is attempted before the beginning or more than one byte beyond the end of the stream.
System.ArgumentOutOfRangeException	<i>offset</i> is greater than the maximum length of the System.IO.MemoryStream .

1
2
3

System.ArgumentException	<i>loc</i> is not a valid System.IO.SeekOrigin value.
System.ObjectDisposedException	The current stream is closed.

1 `MemoryStream.SetLength(System.Int64)`

2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual void SetLength(int64  
5 value)  
6 [C#]  
7 public override void SetLength(long value)
```

8 Summary

9 Sets the length of the current stream to the specified value.

10 Parameters

11
12

Parameter	Description
<code>value</code>	A System.Int64 that specifies the value at which to set the length.

13
14

14 Description

15 If the specified value is less than the current length of the stream, the
16 stream is truncated. If after the truncation the current position within
17 the stream is past the end of the stream, the
18 **System.IO.MemoryStream.ReadByte** method returns -1, the
19 **System.IO.MemoryStream.Read** method reads zero bytes into the
20 provided byte array, and **System.IO.MemoryStream.Write** and
21 **System.IO.MemoryStream.WriteByte** methods append specified
22 bytes at the end of the stream, increasing its length.

23
24
25
26
27
28
29

If the specified value is larger than the current capacity and the stream is resizable, the capacity is increased, and the current position within the stream is unchanged. If the length is increased, the contents of the stream between the old and the new length are initialized to zeros.

30
31
32
33
34
35
36

[*Note:* A **System.IO.MemoryStream** instance must support writing for this method to work. Use the **System.IO.MemoryStream.CanWrite** property to determine whether the current instance supports writing. For additional information, see **System.IO.Stream.CanWrite**.

This method overrides **System.IO.Stream.SetLength**.]

37 Exceptions

38
39

Exception	Condition
<p>System.NotSupportedException</p>	<p>The current stream is not resizable and <i>value</i> is greater than the current System.IO.MemoryStream.Capacity.</p> <p>-or-</p> <p>The current stream does not support writing.</p>
<p>System.ArgumentOutOfRangeException</p>	<p><i>value</i> is negative or is greater than the maximum length of the System.IO.MemoryStream, where the maximum length is (System.Int32.MaxValue - origin), and origin is the index into the underlying buffer at which the stream starts.</p>

- 1
- 2
- 3

1 **MemoryStream.ToArray()** Method

```
2 [ILASM]  
3 .method public hidebysig virtual class System.Byte[]  
4 ToArray()  
  
5 [C#]  
6 public virtual byte[] ToArray()
```

7 **Summary**

8 Writes the entire stream contents to a **System.Byte** array, regardless
9 of the current position within the stream.

10 **Return Value**

11

12 A new **System.Byte** array.

13 **Description**

14 This method returns a copy of the contents of the
15 **System.IO.MemoryStream** as a byte array. If the current instance
16 was constructed on a provided byte array, a copy of the section of the
17 array to which the current instance has access is returned. [*Note:* For
18 additional information, see the **System.IO.MemoryStream**
19 (**System.Byte[]**, **System.Int32**, **System.Int32**) constructor.]

20

21 [*Note:* This method works when the **System.IO.MemoryStream** is
22 closed.]

23 **Behaviors**

24 As described above.

25

MemoryStream.Write(System.Byte[], System.Int32, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void Write(class
System.Byte[] buffer, int32 offset, int32 count)

[C#]
public override void Write(byte[] buffer, int offset, int
count)
```

Summary

Writes a block of bytes to the current stream at the current position using data read from buffer.

Parameters

Parameter	Description
<i>buffer</i>	The System.Byte array to write data from.
<i>offset</i>	A System.Int32 that specifies the zero based byte offset into <i>buffer</i> at which to begin writing from.
<i>count</i>	A System.Int32 that specifies the maximum number of bytes to write from <i>buffer</i> .

Description

If the write operation is successful, the current position within the stream is advanced by the number of bytes written. If an exception occurs, the current position within the stream is unchanged.

Write operations at the end of a resizable **System.IO.MemoryStream** expand the **System.IO.MemoryStream**.

[Note: Use the **System.IO.MemoryStream.CanWrite** method to determine whether the current stream supports writing.]

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

Exceptions

Exception	Condition
System.ArgumentNullException	<i>buffer</i> is null .
System.NotSupportedException	The current stream does not support

	writing. -or- The current position is closer than <i>count</i> bytes to the end of the stream, and the capacity cannot be modified.
System.ArgumentException	(<i>offset</i> + <i>count</i>) is greater than the length of <i>buffer</i> .
System.ArgumentOutOfRangeException	<i>offset</i> or <i>count</i> are negative.
System.IO.IOException	An I/O error occurred.
System.ObjectDisposedException	The current stream is closed.

- 1
- 2
- 3

1 MemoryStream.WriteByte(System.Byte)

2 Method

```
3 [ILASM]  
4 .method public hidebysig virtual void WriteByte(unsigned  
5 int8 value)  
  
6 [C#]  
7 public override void WriteByte(byte value)
```

8 Summary

9 Writes a **System.Byte** to the current stream at the current position.

10 Parameters

11
12

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

13
14

14 Description

15 Write operations at the end of a resizable
16 **System.IO.MemoryStream** expand the
17 **System.IO.MemoryStream**. If the write operation is successful, the
18 current position within the stream is advanced by one byte. If an
19 exception occurs, the position is unchanged.

20
21 [Note: Use the **System.IO.MemoryStream.CanWrite** method to
22 determine whether the current stream supports writing.]

23
24 [Note: This method overrides **System.IO.Stream.WriteByte**.]

25 Exceptions

26
27

Exception	Condition
System.ObjectDisposedException	The current stream is closed.
System.NotSupportedException	The current stream does not support writing.
	-or-
	The current position is at the end of the stream, and the stream's capacity cannot be modified.

1
2
3

1 MemoryStream.WriteTo(System.IO.Stream) Method

```
3 [ILASM]  
4 .method public hidebysig virtual void WriteTo(class  
5 System.IO.Stream stream)  
  
6 [C#]  
7 public virtual void WriteTo(Stream stream)
```

8 Summary

9 Writes the entire contents of the current **System.IO.MemoryStream**
10 instance to a specified stream.

11 Parameters

12
13

Parameter	Description
<i>stream</i>	The System.IO.Stream to write the current memory stream to.

14
15

15 Description

16 [Note: When the current stream is open, this method is equivalent to
17 calling **System.IO.Stream.Write** on the underlying buffer of this
18 stream.]

19 Behaviors

20 As described above.

21 Exceptions

22
23

Exception	Condition
System.ArgumentNullException	<i>stream</i> is null .
System.ObjectDisposedException	The current or target stream is closed.

24
25
26

1 MemoryStream.CanRead Property

```
2 [ILASM]
3 .property bool CanRead { public hidebysig virtual
4 specialname bool get_CanRead() }
5
6 [C#]
7 public override bool CanRead { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether the current stream
9 supports reading.

10 Property Value

11

12 **true** if the current stream supports reading; otherwise **false**.

13 Description

14 This property is read-only.

15

1 MemoryStream.CanSeek Property

```
2 [ILASM]
3 .property bool CanSeek { public hidebysig virtual
4 specialname bool get_CanSeek() }
5
6 [C#]
7 public override bool CanSeek { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether the current stream
9 supports seeking.

10 Property Value

11

12 **true** if the stream supports seeking; otherwise **false**.

13 Description

14 This property is read-only.

15

1 MemoryStream.CanWrite Property

```
2 [ILASM]
3 .property bool CanWrite { public hidebysig virtual
4 specialname bool get_CanWrite() }
5
6 [C#]
7 public override bool CanWrite { get; }
```

7 Summary

8 Gets a **System.Boolean** value indicating whether the current stream
9 supports writing.

10 Property Value

11

12 **true** if the stream supports writing; otherwise, **false**.

13 Description

14 This property is read-only.

15

1 MemoryStream.Capacity Property

```
2 [ILASM]
3 .property int32 Capacity { public hidebysig virtual
4 specialname int32 get_Capacity() public hidebysig virtual
5 specialname void set_Capacity(int32 value) }
6
7 [C#]
8 public virtual int Capacity { get; set; }
```

8 Summary

9 Gets or sets the number of bytes allocated for the current stream.

10 Property Value

11

12 A **System.Int32** containing the number of bytes allocated for the
13 current stream.

14 Description

15 **System.IO.MemoryStream.Capacity** is the buffer length for
16 system-provided byte arrays. If the current stream is created with a
17 specified **System.Byte** array, **System.IO.MemoryStream.Capacity**
18 indicates the length of the portion of the provided array to which the
19 current stream has access. [*Note:* For additional information, see the
20 **System.IO.MemoryStream (System.Byte[], System.Int32,**
21 **System.Int32)** constructor.]

22

23 **System.IO.MemoryStream.Capacity** cannot be set to a value less
24 than the current length of the stream, but can be set to less than the
25 current capacity. If the capacity specified is less than the current
26 capacity, the size of the buffer used to hold the stream may be
27 reduced, but need not be.

28

29 [*Note:* If the value specified for a set operation is less than the default
30 value, for performance reasons the property is set to the default. The
31 default value of the **System.IO.MemoryStream.Capacity** property is
32 unspecified.]

33 Behaviors

34 As described above.

35 Exceptions

36

37

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation

1
2
3

	is negative or less than the current length of the stream.
System.NotSupportedException	A set operation was attempted on a stream that is closed or whose capacity cannot be modified.

1 MemoryStream.Length Property

```
2 [ILASM]
3 .property int64 Length { public hidebysig virtual
4 specialname int64 get_Length() }
5
6 [C#]
7 public override long Length { get; }
```

7 Summary

8 Gets the length of the stream in bytes.

9 Property Value

10

11 A **System.Int64** containing the length of the stream in bytes.

12 Description

13 This property is read-only.

14

15 [*Note:* This property overrides **System.IO.Stream.Length.**]

16 Exceptions

17

18

Exception	Condition
System.ObjectDisposedException	The current stream is closed.

19

20

21

1 MemoryStream.Position Property

```
2 [ILASM]
3 .property int64 Position { public hidebysig virtual
4 specialname int64 get_Position() public hidebysig virtual
5 specialname void set_Position(int64 value) }
6
7 [C#]
8 public override long Position { get; set; }
```

8 Summary

9 Gets or sets the current position within the stream.

10 Property Value

11

12 A **System.Int64** containing the current position within the stream.

13 Description

14 The position cannot be set more than one byte beyond the end of the
15 stream.

16

17 [*Note:* This property overrides **System.IO.Stream.Position**.]

18 Exceptions

19

20

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is negative or greater than one byte beyond the length of the current stream.
System.ObjectDisposedException	The current stream is closed.

21

22