

# 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

# MemoryStream(System.Int32)

## Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(int32
capacity)

[C#]
public MemoryStream(int capacity)
```

### Summary

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

### Parameters

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

### Description

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

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

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

### Exceptions

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

# MemoryStream(System.Byte[])

## Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.Byte[] buffer)

[C#]
public MemoryStream(byte[] buffer)
```

### Summary

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

### Parameters

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

### Description

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

[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 greater than **System.IO.MemoryStream.Capacity**, but the stream can be truncated (see **System.IO.MemoryStream.SetLength**).]

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

### Exceptions

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

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

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

[C#]
public MemoryStream(byte[] buffer, bool writable)
```

## Summary

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

## Parameters

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

## Description

The **System.IO.Stream.CanRead** and **System.IO.Stream.CanSeek** properties of the new instance of the **System.IO.MemoryStream** class are set to **true**. The **System.IO.MemoryStream.Capacity** property is set to the length of the specified **System.Byte** array. 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 **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**.

## Exceptions

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

1  
2  
3

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

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

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

## Summary

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

## Parameters

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

## Description

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

[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**.

## Exceptions

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



1  
2  
3

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

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

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

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

## Summary

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

## Parameters

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

## Description

The **System.IO.MemoryStream.CanRead** and **System.IO.MemoryStream.CanSeek** properties of the new **System.IO.MemoryStream** 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 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 byte buffer, and calls to the **System.IO.MemoryStream.GetBuffer** method throw **System.UnauthorizedAccessException**.

1   **Exceptions**  
2  
3

Exception	Condition
<b>System.ArgumentNullException</b>	<i>buffer</i> is <b>null</b> .
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> or <i>count</i> are negative.
<b>System.ArgumentException</b>	( <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 <b>System.Byte</b> array from which to create the new stream.
<i>index</i>	A <b>System.Int32</b> that specifies the index into <i>buffer</i> at which the stream begins.
<i>count</i>	A <b>System.Int32</b> that specifies the length of the stream in bytes.
<i>writable</i>	A <b>System.Boolean</b> that specifies whether the new stream instance supports writing.
<i>publiclyVisible</i>	A <b>System.Boolean</b> that specifies whether <i>buffer</i> is exposed via the <b>System.IO.MemoryStream.GetBuffer</b> , which returns the <b>System.Byte</b> array from which the stream was created. Specify <b>true</b> to expose <i>buffer</i> ; otherwise, specify <b>false</b> .

## 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
<b>System.ArgumentNullException</b>	<i>buffer</i> is <b>null</b> .
<b>System.ArgumentOutOfRangeException</b>	<i>index</i> or <i>count</i> is negative.
<b>System.ArgumentException</b>	( <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
5 [C#]
6 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

# MemoryStream.GetBuffer() Method

```
[ILASM]
.method public hidebysig virtual class System.Byte[]
GetBuffer()

[C#]
public virtual byte[] GetBuffer()
```

## Summary

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

## Return Value

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

## Description

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

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

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

## Behaviors

As described above.

## Exceptions

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



## Example

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

[C#]

```
using System;
using System.IO;

public class MemoryStreamTest {
    public static void Main() {

        MemoryStream ms = new MemoryStream(10);

        byte[] a = ms.GetBuffer();
        byte[] b = ms.GetBuffer();

        //Force reallocation of the underlying byte array.
        ms.Capacity = 10240;
        byte[] c = ms.GetBuffer();

        if(Object.ReferenceEquals(a, b))
            Console.WriteLine("a and b represent the same
instance.");
        else
            Console.WriteLine("a and b represent the different
instances.");

        if(Object.ReferenceEquals(a, c))
            Console.WriteLine("a and c represent the same
instance.");
        else
            Console.WriteLine("a and c represent the different
instances.");

    }
}
```

The output is

a and b represent the same instance.

- 1 a and c represent the different instances.
- 2
- 3

# MemoryStream.Read(System.Byte[], System.Int32, System.Int32) Method

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

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

## Summary

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

## Parameters

Parameter	Description
<i>buffer</i>	A <b>System.Byte</b> 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 <b>System.Int32</b> that specifies the byte offset in <i>buffer</i> at which to begin writing.
<i>count</i>	A <b>System.Int32</b> that specifies the maximum number of bytes to read.

## Return Value

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

## Description

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

[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.

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

## Exceptions

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

## Example

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

[C#]

```
using System;
using System.IO;

public class MemoryStreamTest {
    public static void Main() {
        byte[] values = new byte [] {0,1,2,3,4,5,6,7,8,9};

        foreach (byte b in values) {
            Console.Write(b);
        }

        Console.WriteLine();

        MemoryStream ms = new MemoryStream (values);

        ms.Read(values, 1, 5);

        foreach (byte b in values) {
            Console.Write(b);
        }
    }
}
```

The output is

0123456789

1	0012346789
2	
3	

# 1 MemoryStream.ReadByte() Method

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

Exception	Condition
System.ObjectDisposedException	The current stream is closed.

# MemoryStream.Seek(System.Int64, System.IO.SeekOrigin) Method

```
[ILASM]
.method public hidebysig virtual int64 Seek(int64 offset,
valuetype System.IO.SeekOrigin loc)

[C#]
public override long Seek(long offset, SeekOrigin loc)
```

## Summary

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

## Parameters

Parameter	Description
<i>offset</i>	A <b>System.Int64</b> 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 <b>System.IO.SeekOrigin</b> value that specifies the seek reference point.

## Return Value

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

## Description

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

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

## Exceptions

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

1  
2  
3

<b>System.ArgumentException</b>	<i>loc</i> is not a valid <b>System.IO.SeekOrigin</b> value.
<b>System.ObjectDisposedException</b>	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

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

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

29  
30 [Note: A **System.IO.MemoryStream** instance must support writing  
31 for this method to work. Use the  
32 **System.IO.MemoryStream.CanWrite** property to determine  
33 whether the current instance supports writing. For additional  
34 information, see **System.IO.Stream.CanWrite**.

35  
36 This method overrides **System.IO.Stream.SetLength**.]

### 37 Exceptions

1  
2  
3

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

# 1 MemoryStream.ToArray() Method

```
2 [ILASM]
3 .method public hidebysig virtual class System.Byte[]
4 ToArray()
5
6 [C#]
7 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

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.]

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

## 23 Behaviors

24 As described above.

# 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 <b>System.Byte</b> array to write data from.
<i>offset</i>	A <b>System.Int32</b> that specifies the zero based byte offset into <i>buffer</i> at which to begin writing from.
<i>count</i>	A <b>System.Int32</b> 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
<b>System.ArgumentNullException</b>	<i>buffer</i> is <b>null</b> .
<b>System.NotSupportedException</b>	The current stream does not support

1  
2  
3

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

# MemoryStream.WriteByte(System.Byte)

## Method

```
[ILASM]
.method public hidebysig virtual void WriteByte(unsigned
int8 value)

[C#]
public override void WriteByte(byte value)
```

### Summary

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

### Parameters

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

### Description

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

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

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

### Exceptions

Exception	Condition
<b>System.ObjectDisposedException</b>	The current stream is closed.
<b>System.NotSupportedException</b>	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

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

```
[ILASM]
.method public hidebysig virtual void WriteTo(class
System.IO.Stream stream)

[C#]
public virtual void WriteTo(Stream stream)
```

## Summary

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

## Parameters

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

## Description

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

## Behaviors

As described above.

## Exceptions

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



# 1 MemoryStream.CanRead Property

```
2 [ILASM]
3 .property bool CanRead { public hidebysig virtual
4 specialname bool get_CanRead() }

5 [C#]
6 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 [C#]  
6 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 [C#]
6 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

# MemoryStream.Capacity Property

```
[ILASM]
.property int32 Capacity { public hidebysig virtual
specialname int32 get_Capacity() public hidebysig virtual
specialname void set_Capacity(int32 value) }

[C#]
public virtual int Capacity { get; set; }
```

## Summary

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

## Property Value

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

## Description

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

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

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

## Behaviors

As described above.

## Exceptions

Exception	Condition
<b>System.ArgumentOutOfRangeException</b>	The value specified for a set operation

1  
2  
3

	is negative or less than the current length of the stream.
<b>System.NotSupportedException</b>	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 [C#]
6 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
<b>System.ObjectDisposedException</b>	The current stream is closed.

19

20

21

# MemoryStream.Position Property

```
[ILASM]
.property int64 Position { public hidebysig virtual
specialname int64 get_Position() public hidebysig virtual
specialname void set_Position(int64 value) }

[C#]
public override long Position { get; set; }
```

## Summary

Gets or sets the current position within the stream.

## Property Value

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

## Description

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

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

## Exceptions

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