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 00]
- *Version:* 2.0.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.

The maximum length of a `System.IO.MemoryStream` is implementation-specific.

[*Note:* Memory streams created with an unsigned byte array provide a non-resizable stream view of the data. When using a byte array, you can neither append to nor shrink the stream, although you might be able to modify the

existing contents depending on the parameters passed into the constructor.]

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
buffer	The `System.Byte` array from which to create the new stream.
index	A `System.Int32` that specifies the index into *buffer* at which the stream begins.
count	A `System.Int32` that specifies the length of the stream in bytes.
writable	A `System.Boolean` that specifies whether the new stream instance supports writing.
publiclyVisible	A `System.Boolean` that specifies whether *buffer* is exposed via `System.IO.MemoryStream.GetBuffer`, which returns the `System.Byte` array from which the stream was created. Specify `true` to expose *buffer*; 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 depending on the value of *writable*, 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`).]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*index* or *count* is negative.
System.ArgumentException	(*index* + *count*) is greater than the length of *buffer*.

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
buffer	The `System.Byte` array from which to create the new stream.
index	A `System.Int32` that specifies the index in *buffer* at which the stream begins.
count	A `System.Int32` that specifies the length of the stream in bytes.
writable	A `System.Boolean` 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 depending on the value of *writable*, 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`.

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*index* or *count* are negative.
System.ArgumentException	(*index* + *count*) is greater than the length of *buffer*.

MemoryStream() Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor()

[C#]
public MemoryStream()
```

Summary

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

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 capacity of the new stream instance can be increased by using the `System.IO.MemoryStream.SetLength` method or by setting the `System.IO.MemoryStream.Capacity` property.

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

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
capacity	A `System.Int32` that specifies the initial size of the internal `System.Byte` 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
System.ArgumentOutOfRangeException	*capacity* 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
buffer	The `System.Byte` 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
System.ArgumentNullException	The *buffer* parameter is `null`.

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
buffer	The `System.Byte` array from which to create the new stream.
writable	A `System.Boolean` 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 depending on the value of *writable*, 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
System.ArgumentNullException	*buffer* is `null`.

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
buffer	The `System.Byte` array from which to create the new stream.
index	A `System.Int32` that specifies the index into *buffer* at which the stream begins.
count	A `System.Int32` 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

System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*index* or *count* is less than zero.
System.ArgumentException	(*index* + *count*) is greater than the length of *buffer*.

MemoryStream.Close() Method

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

[C#]
public override void Close()
```

Summary

Closes the current `System.IO.MemoryStream` instance.

Description

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

The buffer of a closed `System.IO.MemoryStream` is still available, and the `System.IO.MemoryStream.ToArray` and `System.IO.MemoryStream.GetBuffer` methods can be called successfully.

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

MemoryStream.Flush() Method

```
[ILAsm]
.method public hidebysig virtual void Flush()

[C#]
public override void Flush()
```

Summary

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

Description

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

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

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, `System.IO.MemoryStream`(`System.Byte` [], `System.Int32`, `System.Int32`, `System.Boolean`, `true`) 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
System.UnauthorizedAccessException	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.


a and c represent the different instances.
```

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

Return Value

A `System.Int32` that specifies the total number of bytes written 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.

If the read takes place immediately following a seek beyond the end of the stream, the end of the stream is reached.

[*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
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*offset* or *count* is negative.
System.ArgumentException	(*offset* + *count*) is larger than the length of *buffer*.
System.ObjectDisposedException	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


0012346789
```

MemoryStream.ReadByte() Method

```
[ILAsm]
.method public hidebysig virtual int32 ReadByte()

[C#]
public override int ReadByte()
```

Summary

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

Return Value

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

Description

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

If the read takes place immediately following a seek beyond the end of the stream, the end of the stream is reached.

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

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

Changes the position within the current stream by the given offset, which is relative to the stated origin.

Parameters

Parameter	Description
offset	A `System.Int64` that specifies the new position within the stream. This is relative to the *loc* parameter, and can be positive or negative.
loc	A `System.IO.SeekOrigin` 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

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

Exceptions

Exception	Condition
System.IO.IOException	Seeking is attempted before the beginning of the stream.
System.ArgumentOutOfRangeException	*offset* is greater than the maximum length of `System.IO.MemoryStream`.
System.ArgumentException	*loc* is not a valid `System.IO.SeekOrigin` value.
System.ObjectDisposedException	The current stream is closed.

MemoryStream.SetLength(System.Int64) Method

```
[ILAsm]
.method public hidebysig virtual void SetLength(int64 value)

[C#]
public override void SetLength(long value)
```

Summary

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

Parameters

Parameter	Description
value	A `System.Int64` that specifies the value at which to set the length.

Description

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

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.

[*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`.

]

Exceptions

Exception	Condition

System.NotSupportedException	The current stream is not resizable and *value* is greater than the current `System.IO.MemoryStream.Capacity`. -or- The current stream does not support writing.
System.ArgumentOutOfRangeException	*value* is negative or is greater than the maximum length of the `System.IO.MemoryStream` - *origin*, where *origin* is the index into the underlying buffer at which the stream starts.

MemoryStream.ToArray() Method

```
[ILAsm]
.method public hidebysig virtual class System.Byte[] ToArray()

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

Summary

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

Return Value

A new `System.Byte` array.

Description

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

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

Behaviors

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
buffer	The `System.Byte` array to write data from.
offset	A `System.Int32` that specifies the zero based byte offset into *buffer* at which to begin writing from.
count	A `System.Int32` that specifies the maximum number of bytes to write from *buffer*.

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.

If the write takes place immediately following a seek beyond the end of the stream, that stream is zero-byte-extended to the new seek position before the given bytes are written.

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	*buffer* is `null`.
System.NotSupportedException	The current stream does not support writing. -or- The current position is closer than *count* bytes to the end of the stream, and the capacity cannot be modified.
System.ArgumentException	(*offset* + *count*) is greater than the length of *buffer*.
System.ArgumentOutOfRangeException	*offset* or *count* are negative.
System.IO.IOException	An I/O error occurred.
System.ObjectDisposedException	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
value	The `System.Byte` 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.

If the write takes place immediately following a seek beyond the end of the stream, that stream is zero-byte-extended to the new seek position before the given byte is written.

[*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
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.

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
stream	The `System.IO.Stream` to write the current memory stream to.

Description

[*Note:* This method is equivalent to calling *stream*`.Write(this.GetBuffer(), 0, this.GetBuffer().Length)` and passing in the underlying buffer of the current instance.]

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentNullException	*stream* is `null`.
System.ObjectDisposedException	The current or target stream is closed.

MemoryStream.CanRead Property

```
[ILAsm]
.property bool CanRead { public hidebysig virtual specialname bool
get_CanRead() }

[C#]
public override bool CanRead { get; }
```

Summary

Gets a `System.Boolean` value indicating whether the current stream supports reading.

Property Value

`true` if the current stream is open and supports reading; otherwise `false`.

Description

This property is read-only.

MemoryStream.CanSeek Property

```
[ILAsm]
.property bool CanSeek { public hidebysig virtual specialname bool
get_CanSeek() }

[C#]
public override bool CanSeek { get; }
```

Summary

Gets a `System.Boolean` value indicating whether the current stream supports seeking.

Property Value

`true` if the stream is open and supports seeking; otherwise `false`.

Description

This property is read-only.

MemoryStream.CanWrite Property

```
[ILAsm]
.property bool CanWrite { public hidebysig virtual specialname bool
get_CanWrite() }

[C#]
public override bool CanWrite { get; }
```

Summary

Gets a `System.Boolean` value indicating whether the current stream supports writing.

Property Value

`true` if the stream supports writing; otherwise, `false`.

Description

This property is read-only.

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 can 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
System.ArgumentOutOfRangeException	The value specified for a set operation is negative or less than the current length of the stream.
System.NotSupportedException	A set operation was attempted on a stream whose capacity cannot be modified.
System.ObjectDisposedException	The current stream is closed.

MemoryStream.Length Property

```
[ILAsm]
.property int64 Length { public hidebysig virtual specialname int64
get_Length() }

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

Summary

Gets the length of the stream in bytes.

Property Value

A `System.Int64` containing the length of the stream in bytes.

Description

This property is read-only.

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

Exceptions

Exception	Condition
System.ObjectDisposedException	The current stream is closed.

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

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

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The value specified for a set operation is negative or greater than the maximum length of a `System.IO.MemoryStream`.
System.ObjectDisposedException	The current stream is closed.