System.Net.Sockets.NetworkStream Class

```
[ILAsm]
.class public NetworkStream extends System.IO.Stream

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

Assembly Info:

- *Name:* System
- *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

Implements the standard stream mechanism to read and write network data through an instance of the `System.Net.Sockets.Socket` class.

Inherits From: System.IO.Stream

Library: Networking

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.Net.Sockets.NetworkStream` class allows network data to be read and written in the same manner as the `System.IO.Stream` class.

This class supports simultaneous synchronous and asynchronous access to the network data. Random access is not supported and thus the `System.Net.Sockets.NetworkStream.CanSeek` property always returns `false`.

The following properties and methods inherited from the `System.IO.Stream` class are not supported and throw a `System.NotSupportedException` exception when accessed:

- `System.Net.Sockets.NetworkStream.Length`

- `System.Net.Sockets.NetworkStream.Position`

- `System.Net.Sockets.NetworkStream.Seek`

- `System.Net.Sockets.NetworkStream.SetLength`

The `System.Net.Sockets.NetworkStream.Flush` method is reserved for future use but does not throw an exception.

NetworkStream(System.Net.Sockets.Socket, System.IO.FileAccess, System.Boolean) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Net.Sockets.Socket socket, valuetype System.IO.FileAccess
access, bool ownsSocket)

[C#]
public NetworkStream(Socket socket, FileAccess access, bool
ownsSocket)
```

Summary

Constructs and initializes a new instance of the `System.Net.Sockets.NetworkStream` class.

Parameters

Parameter	Description
socket	An instance of the `System.Net.Sockets.Socket` class.
access	One of the values of the `System.IO.FileAccess` enumeration.
ownsSocket	`true` if *socket* is owned by the current instance; otherwise, `false`.

Description

socket is required to be an instance of the `System.Net.Sockets.Socket` class with its `System.Net.Sockets.Socket.Connected` property equal to `true`, `System.Net.Sockets.Socket.Blocking` property equal to `true`, and `System.Net.Sockets.SocketType` equal to `System.Net.Sockets.SocketType.Stream`.

When *ownsSocket* is `true`, the current instance owns *socket*, meaning the `System.Net.Sockets.NetworkStream.Close` and `System.Net.Sockets.NetworkStream.Dispose` methods call the `System.Net.Sockets.Socket.Close` method of *socket*.

Exceptions

Exception	Condition
System.ArgumentNullException	*socket* is `null`.
System.IO.IOException	The `System.Net.Sockets.Socket.Blocking`

	property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.Connected` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.SocketType` property of *socket* is not `System.Net.Sockets.SocketType.Stream`.

NetworkStream(System.Net.Sockets.Socket, System.IO.FileAccess) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Net.Sockets.Socket socket, valuetype System.IO.FileAccess
access)

[C#]
public NetworkStream(Socket socket, FileAccess access)
```

Summary

Constructs and initializes a new instance of the `System.Net.Sockets.NetworkStream` class.

Parameters

Parameter	Description
socket	An instance of the `System.Net.Sockets.Socket` class.
access	One of the values of the `System.IO.FileAccess` enumeration.

Description

This constructor is equivalent to `System.Net.Sockets.NetworkStream.NetworkStream`(*socket*, *access*, `false`).

Exceptions

Exception	Condition
System.ArgumentNullException	*socket* is `null`.
System.IO.IOException	The `System.Net.Sockets.Socket.Blocking` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.Connected` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.SocketType` property of *socket* is not

	`System.Net.Sockets.SocketType.Stream.`

NetworkStream(System.Net.Sockets.Socket) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Net.Sockets.Socket socket)

[C#]
public NetworkStream(Socket socket)
```

Summary

Constructs and initializes a new instance of the `System.Net.Sockets.NetworkStream` class.

Parameters

Parameter	**Description**
socket	An instance of the `System.Net.Sockets.Socket` class.

Description

This constructor is equivalent to `System.Net.Sockets.NetworkStream.NetworkStream`(*socket*, `System.IO.FileAccess.ReadWrite`, `false`).

Exceptions

Exception	**Condition**
System.ArgumentNullException	*socket* is `null`.
System.IO.IOException	The `System.Net.Sockets.Socket.Blocking` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.Connected` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.SocketType` property of *socket* is not `System.Net.Sockets.SocketType.Stream`.

NetworkStream(System.Net.Sockets.Socket, System.Boolean) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Net.Sockets.Socket socket, bool ownsSocket)

[C#]
public NetworkStream(Socket socket, bool ownsSocket)
```

Summary

Constructs and initializes a new instance of the `System.Net.Sockets.NetworkStream` class.

Parameters

Parameter	Description
socket	An instance of the `System.Net.Sockets.Socket` class.
ownsSocket	`true` if *socket* is owned by the current instance; otherwise, `false`.

Description

This constructor is equivalent to `System.Net.Sockets.NetworkStream.NetworkStream`(*socket*, `System.IO.FileAccess.ReadWrite`, *ownsSocket*).

Exceptions

Exception	Condition
System.ArgumentNullException	*socket* is `null`.
System.IO.IOException	The `System.Net.Sockets.Socket.Blocking` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.Connected` property of *socket* is `false`. -or- The `System.Net.Sockets.Socket.SocketType` property of *socket* is not

	`System.Net.Sockets.SocketType.Stream.`

NetworkStream.BeginRead(System.Byte[], System.Int32, System.Int32, System.AsyncCallback, System.Object) Method

```
[ILAsm]
.method public hidebysig virtual class System.IAsyncResult
BeginRead(class System.Byte[] buffer, int32 offset, int32 size,
class System.AsyncCallback callback, object state)

[C#]
public override IAsyncResult BeginRead(byte[] buffer, int offset,
int size, AsyncCallback callback, object state)
```

Summary

Begins an asynchronous operation to read data from the current instance.

Parameters

Parameter	Description
buffer	A `System.Byte` array to store data read from the stream.
offset	A `System.Int32` containing the zero-based position in *buffer* at which to begin storing the data.
size	A `System.Int32` containing the number of bytes to read.
callback	A `System.AsyncCallback` delegate, or `null`.
state	An application-defined object, or `null`.

Return Value

A `System.IAsyncResult` instance that contains information about the asynchronous operation.

Description

To retrieve the results of the operation and release resources allocated by the `System.Net.Sockets.NetworkStream.BeginRead` method, call the `System.Net.Sockets.NetworkStream.EndRead` method, and specify the `System.IAsyncResult` object returned by this method.

[*Note:* The `System.Net.Sockets.NetworkStream.EndRead` method should be called exactly once for each call to the

`System.Net.Sockets.NetworkStream.BeginRead` method.]

If the *callback* parameter is not `null`, the method referenced by *callback* is invoked when the asynchronous operation completes. The `System.IAsyncResult` object returned by this method is passed as the argument to the method referenced by *callback*. The method referenced by *callback* can retrieve the results of the operation by calling the `System.Net.Sockets.NetworkStream.EndRead` method.

The *state* parameter can be any object that the caller wishes to have available for the duration of the asynchronous operation. This object is available via the `System.IAsyncResult.AsyncState` property of the object returned by this method.

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

]

Exceptions

Exception	**Condition**
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*offset* < 0. -or- *offset* > *buffer*.Length. -or- *size* < 0. -or- *size* > *buffer*.Length - *offset*.
System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* Any exception thrown by the `System.Net.Sockets.Socket.BeginReceive` method is caught and rethrown as an `IOException` with the original exception stored in the `System.Exception.InnerException` property.]

System.ObjectDisposedException	The current instance has been disposed.

Example

For an outline of an asynchronous operation, see the `System.Net.Sockets.Socket.BeginAccept` method. For the complete example, see the `System.Net.Sockets.Socket` class overview.

NetworkStream.BeginWrite(System.Byte[], System.Int32, System.Int32, System.AsyncCallback, System.Object) Method

```
[ILAsm]
.method public hidebysig virtual class System.IAsyncResult
BeginWrite(class System.Byte[] buffer, int32 offset, int32 size,
class System.AsyncCallback callback, object state)

[C#]
public override IAsyncResult BeginWrite(byte[] buffer, int offset,
int size, AsyncCallback callback, object state)
```

Summary

Begins an asynchronous operation to write data to the current instance.

Parameters

Parameter	Description
buffer	A `System.Byte` array containing data to write to the stream.
offset	A `System.Int32` containing the zero-based position in *buffer* containing the starting location of the data to write.
size	A `System.Int32` containing the number of bytes to write to the stream.
callback	A `System.AsyncCallback` delegate, or `null`.
state	An application-defined object, or `null`.

Return Value

A `System.IAsyncResult` instance that contains information about the asynchronous operation.

Description

To release resources allocated by the `System.Net.Sockets.NetworkStream.BeginWrite` method, call the `System.Net.Sockets.NetworkStream.EndWrite` method, and specify the `System.IAsyncResult` object returned by this method.

[*Note:* The `System.Net.Sockets.NetworkStream.EndWrite` method should be called exactly once for each call to the `System.Net.Sockets.NetworkStream.BeginWrite` method.]

If the *callback* parameter is not `null`, the method referenced by *callback* is invoked when the asynchronous operation completes. The `System.IAsyncResult` object returned by this method is passed as the argument to the method referenced by *callback*. The method referenced by *callback* can retrieve the results of the operation by calling the `System.Net.Sockets.NetworkStream.EndWrite` method.

The *state* parameter can be any object that the caller wishes to have available for the duration of the asynchronous operation. This object is available via the `System.IAsyncResult.AsyncState` property of the object returned by this method.

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

]

Exceptions

Exception	Condition
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*offset* < 0. -or- *offset* > *buffer*.Length. -or- *size* < 0. -or- *size* > *buffer*.Length - *offset*.
System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* Any exception thrown by the `System.Net.Sockets.Socket.BeginSend` method is caught and rethrown as an `IOException` with the original exception stored in the `System.Exception.InnerException` property.]

System.ObjectDisposedException	The current instance has been disposed.

Example

For an outline of an asynchronous operation, see the `System.Net.Sockets.Socket.BeginAccept` method. For the complete example, see the `System.Net.Sockets.Socket` class overview.

NetworkStream.Close() Method

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

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

Summary

Closes the stream and, if owned by the current instance, the underlying socket.

Description

This method calls `System.Net.Sockets.NetworkStream.Dispose(true )`, which frees both managed and unmanaged resources used by the current instance. When the underlying socket is owned by the current instance, the `System.Net.Sockets.Socket.Close` method of the socket is called, which frees both managed and unmanaged resources used by the socket.

[*Note:* Ownership of a socket is specified using the `System.Net.Sockets.NetworkStream` constructor.

This method overrides `System.IO.Stream.Close`.

]

NetworkStream.Dispose(System.Boolean) Method

[ILAsm]
```
.method family hidebysig virtual void Dispose(bool disposing)
```

[C#]
```
protected virtual void Dispose(bool disposing)
```

Summary

Releases the unmanaged resources used by the current instance and optionally releases the managed resources.

Parameters

Parameter	Description
disposing	A `System.Boolean`. Specify `true` to release both managed and unmanaged resources; specify `false` to release only unmanaged resources.

Description

[*Note:* Ownership of a socket is specified using the `System.Net.Sockets.NetworkStream` constructor.

The `System.Net.Sockets.NetworkStream.Close` method calls this method with the *disposing* parameter set to `true`. The finalizer calls this method with the *disposing* parameter set to `false`.

]

Behaviors

This method closes the current `System.Net.Sockets.NetworkStream` instance releasing all unmanaged resources allocated by the current instance. When the underlying socket is owned by the current instance, the `System.Net.Sockets.Socket.Close` method of the socket is called, which frees the managed and unmanaged resources used by the socket. When the *disposing* parameter is `true`, this method also releases all resources held by any other managed objects allocated by the current instance.

Default

This method closes the current `System.Net.Sockets.NetworkStream` instance releasing all unmanaged resources allocated by the current instance. When the underlying socket is owned by the current instance, the `System.Net.Sockets.Socket.Close` method of the socket is called, which frees the managed and unmanaged resources used by the socket.

How and When to Override

The `System.Net.Sockets.Socket.Dispose` method can be called multiple times by other objects. When overriding this method, do not reference objects that have been previously disposed in an earlier call.

Usage

Use this method to release resources allocated by the current instance.

NetworkStream.EndRead(System.IAsyncResult) Method

```
[ILAsm]
.method public hidebysig virtual int32 EndRead(class
System.IAsyncResult asyncResult)

[C#]
public override int EndRead(IAsyncResult asyncResult)
```

Summary

Ends an asynchronous call to read data from the current instance.

Parameters

Parameter	Description
asyncResult	A `System.IAsyncResult` object that holds the state information for the asynchronous operation.

Return Value

A `System.Int32` containing the number of bytes read from the stream.

Description

This method blocks if the asynchronous operation has not completed.

The `System.Net.Sockets.NetworkStream.EndRead` method completes an asynchronous request that was started with a call to the `System.Net.Sockets.NetworkStream.BeginRead` method. The object specified for the *asyncResult* parameter is required to be the same object as was returned by the `System.Net.Sockets.NetworkStream.BeginRead` method call that began the request.

If the `System.Net.Sockets.NetworkStream.EndRead` method is invoked via the `System.AsyncCallback` delegate specified to the `System.Net.Sockets.NetworkStream.BeginRead` method, the *asyncResult* parameter is the `System.IAsyncResult` argument passed to the delegate's method.

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

]

Exceptions

Exception	Condition
System.ArgumentNullException	*asyncResult* is `null`.
System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* This method catches all exceptions thrown by the `System.Net.Sockets.Socket.EndReceive` method.]
System.ObjectDisposedException	The current instance has been disposed.

Example

For an outline of an asynchronous operation, see the `System.Net.Sockets.Socket.BeginAccept` method. For the complete example, see the `System.Net.Sockets.Socket` class overview.

NetworkStream.EndWrite(System.IAsyncResult) Method

```
[ILAsm]
.method public hidebysig virtual void EndWrite(class
System.IAsyncResult asyncResult)

[C#]
public override void EndWrite(IAsyncResult asyncResult)
```

Summary

Ends an asynchronous call to write data to the current instance.

Parameters

Parameter	Description
asyncResult	A `System.IAsyncResult` object that holds the state information for the asynchronous operation.

Description

This method blocks if the asynchronous operation has not completed.

The `System.Net.Sockets.NetworkStream.EndWrite` method completes an asynchronous request that was started with a call to the `System.Net.Sockets.NetworkStream.BeginWrite` method. The object specified for the *asyncResult* parameter is required to be the same object as was returned by the `System.Net.Sockets.NetworkStream.BeginWrite` method call that began the request.

If the `System.Net.Sockets.NetworkStream.EndWrite` method is invoked via the `System.AsyncCallback` delegate specified to the `System.Net.Sockets.NetworkStream.BeginWrite` method, the *asyncResult* parameter is the `System.IAsyncResult` argument passed to the delegate's method.

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

]

Exceptions

Exception	Condition
System.ArgumentNullException	*asyncResult* is `null`.

System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* This method catches all exceptions thrown by the `System.Net.Sockets.Socket.EndSend` method.]
System.ObjectDisposedException	The current instance has been disposed.

Example

For an outline of an asynchronous operation, see the `System.Net.Sockets.Socket.BeginAccept` method. For the complete example, see the `System.Net.Sockets.Socket` class overview.

NetworkStream.Finalize() Method

```
[ILAsm]
.method family hidebysig virtual void Finalize()

[C#]
~NetworkStream()
```

Summary

Frees unmanaged resources used by the current instance.

Description

[*Note:* Application code does not call this method; it is automatically invoked during garbage collection unless finalization by the garbage collector has been disabled. For more information, see `System.GC.SuppressFinalize`, and `System.Object.Finalize`.

This method calls `System.Net.Sockets.NetworkStream.Dispose`(`false`), which frees unmanaged resources used by the current instance. When the underlying socket is owned by the current instance, it is closed and the managed and unmanaged resources used by the socket are freed.

Ownership of a socket is specified using the `System.Net.Sockets.NetworkStream` constructor.

This method overrides `System.Object.Finalize`.

]

NetworkStream.Flush() Method

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

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

Summary

This method is reserved for future use.

Description

Calling this method does not throw an exception.

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

]

NetworkStream.Read(System.Byte[], System.Int32, System.Int32) Method

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

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

Summary

Reads data from the current instance and stores it in a data buffer.

Parameters

Parameter	**Description**
buffer	A `System.Byte` array to store data read from the stream.
offset	A `System.Int32` containing the zero-based position in *buffer* at which to begin storing the data.
size	A `System.Int32` containing the number of bytes to read.

Return Value

A `System.Int32` containing the number of bytes read from the stream.

Description

When no incoming data is available, this method blocks and waits for data to arrive.

If the remote socket was shut down gracefully (`System.Net.Sockets.Socket.Shutdown` was called on the socket or the `System.Net.Sockets.SocketOptionName.Linger` option was enabled and `System.Net.Sockets.Socket.Close` was called on the socket) and all data was received, this method immediately returns zero.

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

]

Exceptions

Exception	**Condition**

System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*offset* < 0. -or- *offset* > *buffer*.Length. -or- *size* < 0. -or- *size* > *buffer*.Length - *offset*.
System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* This method catches all exceptions thrown by the `System.Net.Sockets.Socket.Receive` method.]
System.ObjectDisposedException	The current instance has been disposed.

NetworkStream.Seek(System.Int64, System.IO.SeekOrigin) Method

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

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

Summary

Throws a `System.NotSupportedException`.

Parameters

Parameter	Description
offset	This parameter is not used.
origin	This parameter is not used.

Description

[*Note:* The `System.IO.Stream` base class uses this method to set the current position in the stream. This functionality is not supported in the `System.Net.Sockets.NetworkStream` class.

This method overrides `System.IO.Stream.Seek`.

]

Exceptions

Exception	Condition
System.NotSupportedException	Any call to this method.

NetworkStream.SetLength(System.Int64) Method

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

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

Summary

Throws a `System.NotSupportedException`.

Parameters

Parameter	Description
value	This parameter is not used.

Description

[*Note:* The `System.IO.Stream` base class uses this method to set the length of the data available on the stream. This functionality is not supported in the `System.Net.Sockets.NetworkStream` class.

This method overrides `System.IO.Stream.SetLength`.

]

Exceptions

Exception	Condition
System.NotSupportedException	Any call to this method.

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

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

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

Summary

Writes data from a specific area of a data buffer to the current instance.

Parameters

Parameter	**Description**
buffer	A `System.Byte` array containing data to write to the stream.
offset	A `System.Int32` containing the zero-based position in *buffer* containing the starting location of the data to write.
size	A `System.Int32` containing the number of bytes to write to the stream.

Description

When no buffer space is available within the underlying protocol, this method blocks unless the socket is in non-blocking mode.

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

]

Exceptions

Exception	**Condition**
System.ArgumentNullException	*buffer* is `null`.
System.ArgumentOutOfRangeException	*offset* < 0. -or- *offset* > *buffer*.Length. -or- *size* < 0.

	-or- *size* > *buffer*.Length - *offset*.
System.IO.IOException	An error occurred while accessing the underlying socket. [*Note:* This method catches all exceptions thrown by the `System.Net.Sockets.Socket.Send` method.]
System.ObjectDisposedException	The current instance has been disposed.

NetworkStream.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` indicates that the current stream supports reading; `false`. indicates that the current stream does not support reading.

Description

This property is read-only.

The value of this property is initially set by the `System.Net.Sockets.NetworkStream` constructors.

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

]

NetworkStream.CanSeek Property

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

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

Summary

Returns the `System.Boolean` value `false` to indicate that the `System.Net.Sockets.NetworkStream` class cannot access a specific location in the data stream.

Property Value

`false`.

Description

This property is read-only.

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

]

NetworkStream.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` indicates that the current stream supports writing; `false` indicates that the current stream does not support writing.

Description

This property is read-only.

The value of this property is set by the `System.Net.Sockets.NetworkStream` constructors.

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

]

NetworkStream.DataAvailable Property

[ILAsm]
```
.property bool DataAvailable { public hidebysig virtual specialname bool get_DataAvailable() }
```

[C#]
```
public virtual bool DataAvailable { get; }
```

Summary

Gets a `System.Boolean` value indicating whether data is available to be read from the underlying socket buffer.

Property Value

`true` indicates that data is available to be read; `false` indicates that there is no data available to be read.

Description

This property is read-only.

Behaviors

As described above.

Default

Accessing this property causes a call to the `System.Net.Sockets.Socket.Available` method of the underlying `System.Net.Sockets.Socket` instance. If the `Available` method returns a non-zero value, indicating data is available to be read, this property returns `true`; otherwise, this property returns `false`.

How and When to Override

Override this property to determine if data is available to be read in the underlying socket buffer.

Exceptions

Exception	Condition
System.ObjectDisposedException	The current instance has been disposed.

NetworkStream.Length Property

[ILAsm]

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

[C#]

```
public override long Length { get; }
```

Summary

Throws a `System.NotSupportedException`.

Description

[*Note:* The `System.IO.Stream` base class implements this property to return the length of the data available on the stream. This functionality is not supported in the `System.Net.Sockets.NetworkStream` class.

This property overrides `System.IO.Stream.Length`.

]

Exceptions

Exception	Condition
System.NotSupportedException	Any attempt to access this property.

NetworkStream.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

Throws a `System.NotSupportedException`.

Description

[*Note:* The `System.IO.Stream` base class implements this property to return or set the current position in the stream. This functionality is not supported in the `System.Net.Sockets.NetworkStream` class.

This property overrides `System.IO.Stream.Position`.

]

Exceptions

Exception	Condition
System.NotSupportedException	Any attempt to access this property.