System.IAsyncResult Interface

```
[ILAsm]
.class interface public abstract IAsyncResult

[C#]
public interface IAsyncResult
```

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)

Summary

Supported by objects that represent the state of an asynchronous operation.

Library: BCL

Description

An object that supports the `System.IAsyncResult` interface stores state information for an asynchronous operation, and provides a synchronization object to allow threads to be signaled when the operation completes.

`System.IAsyncResult` objects are returned by methods that begin asynchronous operations, such as `System.IO.FileStream.BeginRead`, and are passed to methods used to complete asynchronous operations, such as `System.IO.FileStream.EndRead`. `System.IAsyncResult` objects are also passed to methods invoked by `System.AsyncCallback` delegates when an asynchronous operation completes.

IAsyncResult.AsyncState Property

```
[ILAsm]
.property object AsyncState { public hidebysig virtual abstract
specialname object get_AsyncState() }

[C#]
object AsyncState { get; }
```

Summary

Gets the user-provided state object supplied at the time the asynchronous operation was started.

Property Value

The supplied `System.Object`.

Behaviors

The object returned by this property is required to be the object specified as the last parameter to methods that begin asynchronous operations, such as `System.IO.FileStream.BeginRead`.

This property is read-only.

How and When to Override

Implement this property to allow the caller of an asynchronous operation to obtain an application-defined object specified at the start of the operation.

Usage

The object returned by this property can be used to pass state information for the asynchronous operation to a `System.AsyncCallback` delegate.

IAsyncResult.AsyncWaitHandle Property

[ILAsm]

```
.property class System.Threading.WaitHandle AsyncWaitHandle { public
hidebysig virtual abstract specialname class System.Threading.WaitHandle
get_AsyncWaitHandle() }
```

[C#]

```
WaitHandle AsyncWaitHandle { get; }
```

Summary

Gets a `System.Threading.WaitHandle` that can be used to block a thread until an asynchronous operation completes.

Property Value

A `System.Threading.WaitHandle` that is signaled when an asynchronous operation completes.

Behaviors

The object returned by `System.IAsyncResult.AsyncWaitHandle` can be allocated in advance or on demand. However, once allocated it is required to be kept alive until the user calls a method that ends the asynchronous operation, such as `System.IO.FileStream.EndRead`. Only after the operation completes or is canceled, can the object be disposed of.

[*Note:* `WaitHandle` supplies methods that support waiting for synchronization objects to become signaled, such as `System.Threading.WaitHandle.WaitOne`.]

This property is read-only.

Usage

Clients that wait for the operation to complete (as opposed to polling), use this property to obtain a synchronization object to wait on.

IAsyncResult.CompletedSynchronously Property

```
[ILAsm]
.property bool CompletedSynchronously { public hidebysig virtual abstract
specialname bool get_CompletedSynchronously() }

[C#]
bool CompletedSynchronously { get; }
```

Summary

Gets a `System.Boolean` value that specifies whether the asynchronous operation completed synchronously.

Property Value

`true` if the operation synchronously; otherwise `false`.

Behaviors

As described above.

[*Note:* Most implementations of the `System.IAsyncResult` interface will not use this property, and should return `false`.]

This property is read-only.

Usage

Use this property to determine if the asynchronous operation completed synchronously. For example, this property can return `true` for an asynchronous IO operation if the IO request was small.

IAsyncResult.IsCompleted Property

```
[ILAsm]
.property bool IsCompleted { public hidebysig virtual abstract specialname
bool get_IsCompleted() }

[C#]
bool IsCompleted { get; }
```

Summary

Gets a `System.Boolean` value that specifies whether an asynchronous operation has completed.

Property Value

`true` if the operation has completed; otherwise `false`.

Behaviors

As described above.

This property is read-only.

Usage

Clients that poll for operation status (as opposed to waiting on a synchronization object) use this property to determine the status of the operation.