

1 System.IAsyncResult Interface

2
3

```
4 [ILASM]  
5 .class interface public abstract IAsyncResult  
6 [C#]  
7 public interface IAsyncResult
```

8 Assembly Info:

- 9 • Name: mscorlib
- 10 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 11 • Version: 1.0.x.x
- 12 • Attributes:
 - 13 ○ CLSCompliantAttribute(true)

14 Summary

15

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

18 **Library:** BCL

19

20 Description

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

25

26 **System.IAsyncResult** objects are returned by methods that begin
27 asynchronous operations, such as
28 **System.IO.FileStream.BeginRead**, and are passed to methods used
29 to complete asynchronous operations, such as
30 **System.IO.FileStream.EndRead**. **System.IAsyncResult** objects
31 are also passed to methods invoked by **System.AsyncCallback**
32 delegates when an asynchronous operation completes.

33

1 | AsyncResult.AsyncState Property

```
2 | [ILASM]  
3 | .property object AsyncState { public hidebysig virtual  
4 | abstract specialname object get_AsyncState() }  
5 |  
6 | [C#]  
7 | object AsyncState { get; }
```

7 | Summary

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

10 | Property Value

11

12 | The supplied **System.Object**.

13 | Behaviors

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

17

18 | This property is read-only.

19 | How and When to Override

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

23 | Usage

24 | The object returned by this property can be used to pass state
25 | information for the asynchronous operation to a
26 | **System.AsyncCallback** delegate.

27

1 IAsyncResult.AsyncWaitHandle Property

```
2 [ILASM]
3 .property class System.Threading.WaitHandle AsyncWaitHandle
4 { public hidebysig virtual abstract specialname class
5 System.Threading.WaitHandle get_AsyncWaitHandle() }
6
7 [C#]
8 WaitHandle AsyncWaitHandle { get; }
```

8 Summary

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

11 Property Value

12

13 A **System.Threading.WaitHandle** that is signaled when an
14 asynchronous operation completes.

15 Behaviors

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

22

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

26

27 This property is read-only.

28 Usage

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

31

1 **IAsyncResult.CompletedSynchronously** 2 **Property**

```
3 [ILASM]  
4 .property bool CompletedSynchronously { public hidebyref  
5 virtual abstract specialname bool  
6 get_CompletedSynchronously() }  
  
7 [C#]  
8 bool CompletedSynchronously { get; }
```

9 **Summary**

10 Gets a **System.Boolean** value that specifies whether the
11 asynchronous operation completed synchronously.

12 **Property Value**

13

14 **true** if the operation synchronously; otherwise **false**.

15 **Behaviors**

16 As described above.

17

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

20

21 This property is read-only.

22 **Usage**

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

26

1 IAsyncResult.IsCompleted Property

```
2 [ILASM]  
3 .property bool IsCompleted { public hidebysig virtual  
4 abstract specialname bool get_IsCompleted() }  
5 [C#]  
6 bool IsCompleted { get; }
```

7 Summary

8 Gets a **System.Boolean** value that specifies whether an asynchronous
9 operation has completed.

10 Property Value

11

12 **true** if the operation has completed; otherwise **false**.

13 Behaviors

14 As described above.

15

16 This property is read-only.

17 Usage

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

21