

This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. [Learn more](#)



Microsoft Logo



Gray Pipe

[Developer Network](#) [Developer Network](#) [Developer](#)

[Subscriber portal](#)

[Get tools](#)

- [Downloads](#)
- [Programs](#)
- [Community](#)
- [Documentation](#)



search clear

[Programming with the .NET Framework Including Asynchronous Calls](#)

[Asynchronous Programming Design Pattern](#)

[Asynchronous Programming Design Pattern](#) [Asynchronous Method Signatures](#)

[Asynchronous Method Signatures](#)

[Asynchronous Design Pattern Overview](#)

[Asynchronous Method Signatures](#)

[IAsyncResult Interface](#)

[AsyncCallback Delegate for Asynchronous Operations](#)

[TOC](#)

This documentation is archived and is not being maintained.

Asynchronous Method Signatures

.NET Framework 1.1

The method signature is the ordering of the parameter types in a method. The rules for the **BeginInvoke** asynchronous method signature are:

- Include all *IN* parameters.
- Include all *OUT* parameters.
- Include all *IN/OUT* parameters.
- Include all *ByRef* parameters.

- Take **AsyncCallback** and **AsyncState** (available through the **AsyncState** property on the **IAsyncResult** interface) as the last two parameters.
- Return **IAsyncResult**.

The rules for the **EndInvoke** asynchronous method signature are:

- Include all *IN/OUT* parameters.
- Include all *OUT* parameters.
- Include all *ByRef* parameters.
- Take **IAsyncResult** as last parameter.
- Return the original return type from the original method signature.

The result object (**IAsyncResult**) is returned from the begin operation, and can be used to obtain status on whether the asynchronous begin operation has completed. The result object is passed to the end operation, which returns the final return value of the call. An optional callback can be supplied in the begin operation. If the callback is supplied, when the call completes, the callback will be called; and code in the callback can call the end operation.

See Also

[Asynchronous Design Pattern Overview](#) | [IAsyncResult Interface](#) | [AsyncCallback Delegate for Asynchronous Operations](#) | [Asynchronous Programming](#)

Show: Inherited Protected

Dev centers

- [Windows](#)
- [Office](#)
- [Visual Studio](#)
- [Microsoft Azure](#)
- [More...](#)

Learning resources

- [Microsoft Virtual Academy](#)
- [Channel 9](#)
- [MSDN Magazine](#)

Community

- [Forums](#)
- [Blogs](#)
- [Codeplex](#)

Support


- [Self support](#)

Programs

- [BizSpark \(for startups\)](#)
- [Microsoft Imagine \(for students\)](#)

[United States \(English\)](#)

- [Newsletter](#)
- [Privacy & cookies](#)
- [Terms of use](#)
- [Trademarks](#)

logo

© 2017 Microsoft

© 2017 Microsoft