**What's New for Visual C# in Visual Studio 2012**

**Visual Studio 2012**

This page lists the new and enhanced features provided for Visual C# in Visual Studio 2012.

For information about how to create Windows Store apps by using Visual Studio 2012, see [Develop Windows Store apps using Visual Studio 2012](http://go.microsoft.com/fwlink/p/?LinkId=248364).

[Async and Await](javascript:void(0))

You can write asynchronous code more easily and intuitively by using the Async feature, which makes asynchronous programming almost as straightforward as synchronous programming.

Previously, writing asynchronous code has required you to define callbacks (also referred to as continuations) to capture what occurs after an asynchronous process finishes. This complicates your code and makes routine tasks, such as exception handling, awkward and difficult. If you use the Async feature, the compiler does most of the work for you.

The feature introduces two keywords, [async](http://msdn.microsoft.com/en-us/library/hh156513(v=vs.110).aspx) and [await](http://msdn.microsoft.com/en-us/library/hh156528(v=vs.110).aspx). The async modifier specifies that a method is an async method. An async method returns a [Task](http://msdn.microsoft.com/en-us/library/system.threading.tasks.task(v=vs.110).aspx) or [Task<TResult>](http://msdn.microsoft.com/en-us/library/dd321424(v=vs.110).aspx) that represents the ongoing work of the method. The task contains information that the caller of the asynchronous method can use, such as the status of the task, its unique ID, and the method's result.

Typically, an await operator is applied to the returned task in an await expression. The await expression suspends execution of the method that contains it until the awaited task is complete. In the meantime, control is returned to the caller of the suspended method.

For more information, see [Asynchronous Programming with Async and Await (C# and Visual Basic)](http://msdn.microsoft.com/en-us/library/hh191443(v=vs.110).aspx).

[Caller Information](javascript:void(0))

This version of Visual C# makes it easier to obtain information about the caller of a method. By using Caller Info attributes, you can identify the file path of the source code, the line number in the source code, and the member name of the caller. This information is helpful for tracing, for debugging, and for creating diagnostic tools.

For more information, see [Caller Information (C# and Visual Basic)](http://msdn.microsoft.com/en-us/library/hh534540(v=vs.110).aspx).