Visual Basic Programming Guide

Visual Studio 2015

As with any modern programming language, Visual Basic supports many common programming constructs and language elements. This guide describes all the major elements of programming with Visual Basic.

In This Section

Program Structure and Code Conventions (Visual Basic)
: Contains documentation on the basic structure and code conventions of Visual Basic, such as naming conventions, comments in code, and limitations within Visual Basic.

Visual Basic Language Features
: Provides links to topics that introduce and discuss important features of Visual Basic, including LINQ and XML literals.

COM Interop (Visual Basic)
: Explains the interoperability issues associated with creating and using component object model (COM) objects with Visual Basic.

Related Sections

Visual Basic Language Reference
: Provides reference information about various aspects of Visual Basic programming.

Visual Basic Command-Line Compiler
: Offers information on using the Visual Basic command-line compiler, the compiler options, and the Keyword Upgrade tool.

JANUARY 2015 VOLUME 30 NUMBER 1

Visual Basic .NET - 14 Top Improvements in Visual Basic 14

By Lucian Wischik | January 2015

Visual Basic 14 is the newest version of Visual Basic, and it will ship as part of Visual Studio 2015. This version has been rewritten from scratch in about 1.3 million lines of VB code—earlier versions were actually written in C++. The team took advantage of the rewrite to rethink every part of VB from the ground up. I asked the team to pick out their top 14 improvements. They chose their favorites from across the board—in the coding experience, in the project system fundamentals and in the language itself.

Better Coding Experience

1. Refactorings "Just the fact that we finally get refactoring built directly into the product is huge."—.NET MVP Jim Wooley

It used to be you had to buy add-on products just to get essential refactorings for Visual Basic, like extracting a method or inlining a temporary variable. There were a few refactorings for C#, and Microsoft had partnered with Developer Express to make its Refactor! add-in available to Visual Basic users since Visual Studio 2005. Now refactorings are built into Visual Studio 2015! To use them, click on an identifier, or highlight a sub-expression. Then hit Ctrl+Dot, or right-click and choose Quick Actions. This brings up a light bulb context menu of relevant actions, as shown in **Figure 1**.

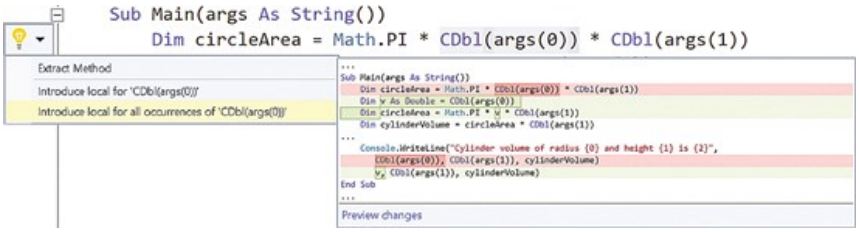


Figure 1 Visual Basic 14 Now Has Built-in Refactoring

Note that the refactorings are context-aware. For instance, if you extract the right-hand side of "Dim circleArea = Math.PI * radius * radius" out into a method, Visual Basic suggests the name "GetCircleArea" for that method. And it puts you into inline-rename mode if you want to change the name further. What's smart about this inline-rename is that it can detect and warn you about name clashes if you pick a name that's already in use, it can avoid clashes where possible, and it works across your entire solution, even changing names in C# projects, as well.

2. Analyzers "The feature is spectacular. I have so many ideas for how to use this feature ... all the little things I see in people's code."—Overheard from a Windows PowerShell MVP

Analyzers are a way to put those light bulbs, code actions and error squiggles into your own hands. You can use them to enforce coding guidelines throughout your team. If you're flown in to debug a problem, you can plug in an analyzer to quickly find common code defects in the entire solution. And many of the libraries you use can become "code-aware" with their own built-in analyzers. For instance, suppose you haven't yet used the Microsoft Azure Storage library or haven't read articles on best practices. Because the library now comes with an analyzer that detects common pitfalls in use of its APIs, you can be immediately confident that you're using it properly. It's like having an expert code reviewer stand over your shoulder as you type.

You can add analyzers to your project under the new References | Analyzers node, (or via NuGet). Once there they become part of your project's compilation—they run live as you type to show live error squiggles. They run when you build your project in Visual Studio or on the command line, and will even run on build servers. Analyzers get a chance to "crack open" the internals of the compiler, to look at the syntax trees of the project's source code and its types and members. Developers are pleasantly surprised to discover how easy it is to code their expert domain knowledge into analyzers, thanks to these syntax trees and types+members. My favorite analyzer is one that detects where my team has been using Async Sub methods that should have been Async Function … As Task and issues a warning. This is a rough corner of asynchronous programming that not enough people are aware of, and it leads to difficult-to-catch concurrency bugs, so it's great that my team can now catch the error at compile time. To get started writing your own analyzers, go to roslyn.codeplex.com.

3. No Need to Cursor off the Line "We do the right thing now."—Dustin Campbell, Visual Basic Team Member

As a VB user, you've long been used to typing some code, then doing a quick "down-then-up cursor" to see if any error squiggles appear. Or you'd write code to fix an error squiggle, but then have to do the "down-then-up" for the squiggle to disappear.

Now you don't have to go through all of that. Just leave the cursor where it is, and error squiggles will appear or disappear themselves.

4. References in XML Doc Comments "For people passionate about docs, this is an enormous step in the right direction."—.NET MVP Sam Harwell

Are you passionate about XML doc comments? Here's a small example:

```
''' <summary>
''' Similar to <see cref="List(Of Integer).Count"/>
''' </summary>
''' <param name="e">Thing to count</param>
''' <remarks></remarks>
Sub Count(e As IEnumerable)
End Sub
```

In previous versions of VB, when you typed out cref and param-name in your comments, you got completion-list help, but beyond that you were on your own. The compiler did some minimal validation to check that the names

existed, but they were typeset in grey and not easy to find, look up or refactor.

Now in Visual Basic 14 the cref and param-name arguments are colorized properly. You can hover over them to see tooltips. When you do a rename-symbol refactoring (Ctrl+R, Ctrl+R), Visual Basic renames all references to a symbol, including those in cref and param-name. You can right-click on one of them and Go to Definition, or Find All References. If you want to refer to a method that has several overloads, you can now unambiguously refer to the single overload you want. These changes all make it easier to type references in your XML doc-comments—and get them right.

Project System Fundamentals

5. References Node in Solution Explorer "Fact is, we all have to tweak references daily."—Lucian Wischik, Visual Basic Team Member

Figure 2 shows how a typical Visual Basic 14 project looks in Solution Explorer.

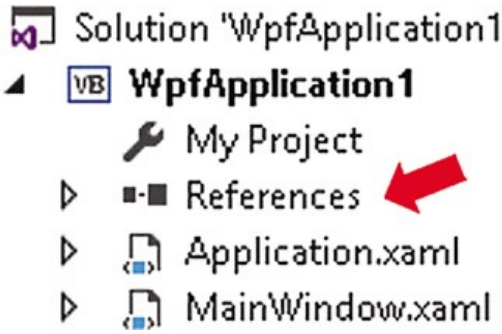


Figure 2 The References Node Is Now Shown in Solution Explorer

What's new is the References node. This used to be hidden and you had to click Show All Files to see it—but that also showed lots of irrelevant files.

This previous behavior might have made sense 10 years ago when you'd start with a Windows Forms project and it would generally have the right set of references. But it's a reality of modern development nowadays that the References node is used frequently—especially to manage NuGet references. It's a small but handy convenience to be able to find it easily in Solution Explorer.

6. Shared Projects "This is really just nice tooling on top of linked files that makes them easier to work with."—Windows Developer MVP Morten Nielsen

Suppose you want to share code between two or more projects. It's a common-enough situation, for example when maintaining both Windows Presentation Foundation (WPF) and Windows Phone versions of an app. The goal is always the same: maximize code reuse, so, for example, a bug fix you make for one project will automatically benefit the other project.

In the past, you could choose between two techniques: use linked files to share common source code, or rearchitect your shared code into a Portable Class Library to share a common binary. Now Visual Basic 14 allows a third, powerful technique: Shared Projects.

Why would you use Shared Projects? The task of sharing code is deep and challenging with no good "one-size-fits–all" solution. Portable Class Libraries are a good, clean solution, but they force you to architect your code so the

common code never calls into the WPF or Phone projects; it only calls into system APIs that are present on both WPF and Phone. Shared Projects are easier to use because they don't require this rearchitecting.

To set up a shared project, right-click on your solution and select Add | New Project | VB | Shared Project. Next, right-click on each project's Reference node in turn and choose Add | Shared Projects. A Shared Project is just a collection of the source files, XAML files, images and other assets that will be included in each project that references it.

For each of your projects you can also set up My Project | Compile | Advanced Compile Options | Custom Constants with custom constants—WPF and PHONE, for example. Then, in your shared code, you can call into project-specific APIs like this:

```
#If WPF Then
  ' nothing needed
#ElseIf PHONE Then
  ShowBatteryStatus()
#End If
```

7. 50 Percent Faster Compile Times "50 percent is no joke."—.NET MVP Sam Harwell

The Visual Basic compiler used to be written in C++. For Visual Basic 14, the team has rewritten it completely in VB, which has made it considerably faster! Here are two comparisons:

- A large solution build (1.3 million lines of code) went from 68 seconds to 41 seconds.
- A cold solution load (a Windows Store app) went from 6.7 seconds to 4.6 seconds

That's a considerable time saving. It's great when your mind can stay on track rather than meandering in the delay between "finish coding" and "press F5 to hit a breakpoint."

The 50 percent performance increase might come as a surprise to people who think that C++ is faster than VB. The truth is that algorithms, data structures and concurrency are where you get real speed benefits. The performance boost from rewriting in VB comes from many places: rethinking the data structures; being able to express algorithms more cleanly and refactor more safely; using async and the threadpool; using the Visual Studio profiling tools to discover CPU and memory-allocation hotspots; and using analyzers to detect simple .NET performance gotchas like unnecessary boxing.

8. Lambdas and LINQ Expressions in a Watch Window "Awesome!"—Visual Studio Uservoice User Marco Senn-Haag

LINQ and lambdas are a great way to summarize data. One of the places this is needed most is at debug time, in the Watch and Immediate windows. In the past, any use of LINQ or lambdas here generated an error:

```
Evaluation of lambda expressions is not valid in the debugger.
```

Now, as you can see in **Figure 3**, it just works! For instance, if you're at a breakpoint where you have a collection called "customers," you can get a quick read on the collection by writing this in the watch window:

```
From c In customers Where c.OrderStatus = "Unfulfilled" Select c.LastName
```

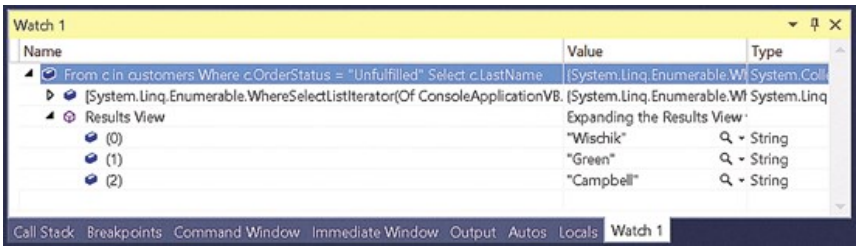


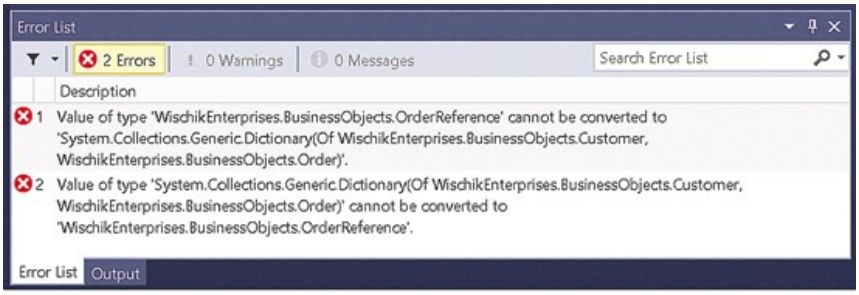
Figure 3 Lambdas and LINQ Expressions in the Watch Window

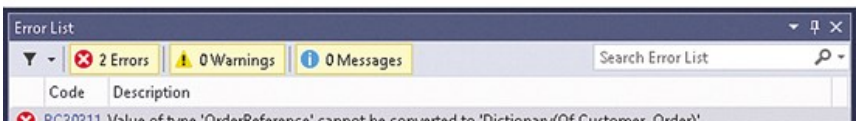
Did you know you can use the Immediate Window without even launching your program? For instance, if you've just written a module with a function GetName, you can open the Immediate window (Debug | Windows | Immediate) and type "? GetName()" and it will evaluate.

Visual Studio 2015 will also have better support for Edit and Continue such as in Async and Iterator methods, as well as in more common situations like inside of and around LINQ queries and lambdas, even allowing you to add a new query or lambda expression to an existing method. Although this didn't make it into the Visual Studio 2015 Preview, you'll be able to do all of this in the final release.

9. Better Error List "The before and after really tell a great story."—Visual Basic Team Member Anthony D. Green

The Error List in Visual Basic 14 has numerous practical improvements, ones that answer long-standing user requests (see **Figure 4**). Before, the Error List used to show fully qualified type names; now it shows only minimally qualified type names so you can more easily read the error message. And it shows the error code, as well, which is handy for sorting by error code. Even better, the error code is a hyperlink to an Internet search, often more useful than a link to the MSDN documentation page. Also, you can filter each column in the error list like you do in Excel.





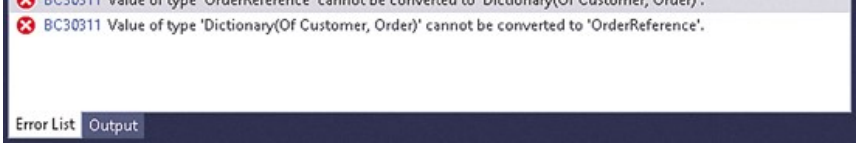


Figure 4 The Error List in Visual Studio 2015 (bottom) is more Readable and Versatile Than in Visual Studio 2013 (top)

Sometimes when making big changes it's easy to get your solution into a state where a lot of downstream code is broken. In the past, VB would show only the first 101 errors. This made it difficult to get an idea of just how widespread the break was, or to get an overview of what kinds of changes you'd have to make. Now, Visual Basic 14 shows all errors without limit. (This can be configured under Tools | Options | Text Editor | Basic | Advanced | Show diagnostics for closed files.)

Language Improvements

10. Null-Propagating Operators "The constant need for null checking bites everyone both in productivity and in bugs."—.NET MVP Deborah Kurata

Suppose you have a Customer class with an Address field that might legitimately be null, perhaps because your app doesn't require addresses to be typed in. Previously, any code that needed to do something with the address—like display it to the user—would need to be safeguarded with null checking to take this into account. Such null checks quickly get tedious. With Visual Basic 14 you can elegantly handle the possibility of a null like this, using the new ?. operator:

```
Console.WriteLine("{0} ({1})",
  customer.Name,
  customer.Address?.Country)
```

The ?. operator is simply shorthand for the common but cumbersome pattern of assigning to a temporary variable and then checking for null:

```
Dim _temp = customer.Address
Console.WriteLine("{0} ({1})",
  customer.Name,
  If(_temp Is Nothing, Nothing, _temp.Country))
```

You can use ?. in a sequence and mix it with the regular dot operator, a?.b.c?.d, for example. It reads left-to-right. Any null value used by ?. will just stop the sequence short and give the answer Nothing, and any null value used by . will throw a NullReferenceException as usual.

The ?. operator is a null-conditional version of the . operator. There are null-conditional versions of most other operators, as well: indexing, array?(i); delegate invocation, delegate?(args); dictionary lookup, dict?!key; and XML axis properties, xml?.@attr, xml?.<key>, xml?...<key>.

You can use ?. in other handy ways, too:

```
If customer?.Age > 50 Then ...
' If branch taken only if customer is non-null AND is older than 50
Dim name = If(customer?.Name, "blank")
' Pick a default name if customer is null
Dim first = customers?.FirstOrDefault()
' Only invoke this method if customers is non-null
```

11. Multiline String Literals "Is it sad that I'm so excited for multiline string literals?"—Forum User Scaramouche

This is how you used to write multiline strings, peppered with vbCrLfs:

```
Dim json = "{" & vbCrLf &
"  'Name': 'Bad Boys'," & vbCrLf &
"  'ReleaseDate': '1995-4-7T00:00:00'," & vbCrLf &
"  'Genres': ['Action','Comedy']" & vbCrLf &
"}"
```

Understandably, a common request has been to allow string literals that span multiple lines. Now, in Visual Basic 14, you can:

```
Dim json = "{
  'Name': 'Bad Boys',
  'ReleaseDate': '1995-4-7T00:00:00',
  'Genres': ['Action','Comedy']
}"
```

A related noteworthy feature—also commonly requested—is that you can now put comments within multiline statements. Previously, they weren't allowed inside LINQ expressions like the following:

```
Dim q = From x In y ' This is a from clause
        Where x < z ' And this is the where
        Select x    ' This select is redundant
```

12. String Interpolation "String interpolation makes for much simpler code, and makes the intent pop out."—Channel9 User Judah

String interpolation is an easier way of writing strings with expressions in them, like so:

```
Dim s = $"hello {p.Name} you are {p.Height:0.00}m tall"
```

This is simply shorthand for the following:

```
Dim s = String.Format("hello {0} you are {1:0.00}m tall", p.Name, p.Height)
```

String interpolation is often easier to write than an explicit call to String.Format because it saves you having to juggle the positional placeholders {0} and {1}. And, of course, there's full colorization and IntelliSense for the expressions inside the holes. String interpolation works particularly well with programmatic strings, as in these examples:

```
Dim fn = $"C:\Documents\{folder}\{file}.{ext}"
Dim url = $"http://{site}/{path}/{file}?search={query}"
```

As is normal for String.Format, this formats the string using the current culture. If you're constructing a programmatic string that includes a floating point number, such as when you're passing latitude and longitude to a Web service, you'll most likely want InvariantCulture instead. This will be supported in Visual Studio 2015, but at press time the design wasn't yet settled.

Note that string interpolation isn't yet in Visual Studio 2015 Preview, but will be present in Visual Studio 2015 before it's final.

13. NameOf "This will be such a help for so many scenarios I can think of."—Roslyn Forum Member ewwloyd

The NameOf operator is a better way of embedding a string literal in your code, when that string literal refers to a name in your source code. Here's one example:

```
Sub f(s As String)
  If s Is Nothing Then Throw New ArgumentNullException(NameOf(s))
End Sub
```

The NameOf operator isn't evaluated at run time: It's a compile-time constant, in this case the constant string "s." The reason to use NameOf(s) is that it safeguards you against typos. For instance, if you rename the method parameter, then the NameOf argument will be renamed automatically. That wouldn't have happened with just a string literal. Here's another place where NameOf works well:

```
Private _age As Integer
Property Age As Integer
  Get
    Return _age
  End Get
```

```
    Set
      _age = Value
      RaiseEvent PropertyChanged(
        Me, New PropertyChangedEventArgs(NameOf(Age)))
    End Set
  End Property
```

Note that NameOf isn't yet in Visual Studio 2015 Preview, but will be present in Visual Studio 2015 before it's final.

14. Open Source "We're trying to engage the community. There are a lot of smart people out there. We'll look at pull requests from the community just like we do our own ideas."—C#/Visual Basic Architect Anders Hejlsberg

The final improvement is not in Visual Basic itself, but in the process of working with VB.

The source code of the VB compiler is now open source. So is the design process of the language itself. Each new feature proposal is made in the open, with full public scrutiny. The members of the Microsoft Visual Basic Language Design Team are essentially now stewards of the language. The team looks at proposals, considers them deeply, sees if there are unexpected gotchas or corner-cases, and determines if they meet the bar for inclusion in the language. The Language Design Meeting minutes are published openly. It's truly an exciting time to be on the Visual Basic Language Design Team, and an exciting time to be a user of VB.

Wrapping Up

There are a lot of improvements in Visual Basic 14. This article has covered almost half of them. The general theme has been to make the existing VB work better in easy-to-use ways, without introducing difficult new concepts. For more information, check out roslyn.codeplex.com and blogs.msdn.com/vbteam.

This article refers to prerelease versions of Visual Basic 14 and Visual Studio 2015. All information is subject to change.

Lucian Wischik *is on the Visual Basic/C# Language Design Team at Microsoft, with particular responsibility for VB. Before joining Microsoft he worked in academia on concurrency theory and async. He's a keen sailor and long-distance swimmer. Reach him at lwischik@microsoft.com.*

Thanks to the following Microsoft technical experts for reviewing this article: Dustin Campbell and Anthony D. Green

Dustin Campbell is a principal program manager on the Visual Studio team, where he works on the Visual Basic and C# IDE experiences. For the last five years, he has been focused on rebuilding the VB and C# compilers and IDEs as part of Project Roslyn.

Anthony D. Green is a program manager on the VB/C#/F# team at Microsoft who has been rewriting the VB & C# compilers (in VB and C#). Anthony has been programming in BASIC programming languages since age 14, mostly in QBasic and VB, and has been an impassioned member of the VB.NET community since 2004.

.NET Framework Development Guide

.NET Framework (current version)

This section explains how to create, configure, debug, secure, and deploy your .NET Framework apps. The section also provides information about technology areas such as dynamic programming, interoperability, extensibility, memory management, and threading.

In This Section

.NET Framework Application Essentials
: Provides information about basic app development tasks, such as programming with app domains and assemblies, using attributes, formatting and parsing base types, using collections, handling events and exceptions, using files and data streams, and using generics.

Data and Modeling in the .NET Framework
: Provides information about how to access data using ADO.NET, Language Integrated Query (LINQ), WCF Data Services, and XML.

Developing Client Applications with the .NET Framework
: Explains how to create Windows-based apps by using Windows Presentation Foundation (WPF) or Windows Forms.

Developing Web Applications with ASP.NET
: Provides links to information about using ASP.NET to build enterprise-class web apps with a minimum of coding.

Developing Service-Oriented Applications with WCF
: Describes how to use Windows Communication Foundation (WCF) to build service-oriented apps that are secure and reliable.

Developing Windows Service Applications
: Explains how you can use Visual Studio and the .NET Framework to create an app that is installed as a service, and start, stop, and otherwise control its behavior.

Parallel Processing and Concurrency in the .NET Framework
: Provides information about managed threading, parallel programming, and asynchronous programming design patterns.

Network Programming in the .NET Framework
: Describes the layered, extensible, and managed implementation of Internet services that you can quickly and easily integrate into your apps.

Configuring .NET Framework Apps
: Explains how you can use configuration files to change settings without having to recompile your .NET Framework apps.

Compiling Apps with .NET Native
: Explains how you can use the .NET Native precompilation technology to build and deploy Windows Store apps. .NET Native compiles apps that are written in managed code (C#) and that target the .NET Framework to native code.

Security in the .NET Framework

Provides information about the classes and services in the .NET Framework that facilitate secure app development.

Debugging, Tracing, and Profiling
: Explains how to test, optimize, and profile .NET Framework apps and the app environment. This section includes information for administrators as well as developers.

Developing for Multiple Platforms with the .NET Framework
: Provides information about how you can use the .NET Framework to build assemblies that can be shared across multiple platforms and multiple devices such as phones, desktop, and web.

Deploying the .NET Framework and Applications
: Explains how to package and distribute your .NET Framework app, and includes deployment guides for both developers and administrators.

.NET Framework Performance
: Provides information about caching, lazy initialization, reliability, and ETW events.

Building Workflows in the .NET Framework
: Provides information about the programming model, samples, and tools for using Windows Workflow Foundation (WF).

Advanced Reading for the .NET Framework
: Provides information about advanced development tasks and techniques in the .NET Framework, including extensibility, interoperability, and reflection. Also includes the reference topics for unmanaged APIs that can be used by managed apps, such as runtime hosts, compilers, disassemblers, debuggers, and profilers.

Reference

.NET Framework Class Library
: Supplies syntax, code examples, and usage information for each class that is contained in the .NET Framework namespaces.

Related Sections

Getting Started with the .NET Framework
: Provides a comprehensive overview of the .NET Framework and links to additional resources.

What's New in the .NET Framework
: Describes key new features and changes in the latest version of the .NET Framework. Includes lists of new and obsolete types and members, and provides a guide for migrating your apps from the previous version of the .NET Framework.

.NET Framework Tools
: Describes the tools that help you develop, configure, and deploy apps by using .NET Framework technologies.

.NET Framework Samples
: Provides links to the MSDN Code Samples Gallery for sample apps that demonstrate .NET Framework technologies.

ASP.NET Overview

ASP.NET is a unified Web development model that includes the services necessary for you to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications you have access to classes in the .NET Framework. You can code your applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic and C#. These languages enable you to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

If you want to try ASP.NET, you can install Visual Web Developer Express using the Microsoft Web Platform Installer, which is a free tool that makes it simple to download, install, and service components of the Microsoft Web Platform. These components include Visual Web Developer Express, Internet Information Services (IIS), SQL Server Express, and the .NET Framework. All of these are tools that you use to create ASP.NET Web applications. You can also use the Microsoft Web Platform Installer to install open-source ASP.NET and PHP Web applications.

This topic describes the following features of ASP.NET and of Visual Web Developer, the development environment for creating ASP.NET applications.

The Three Flavors of ASP.NET: Web Forms, MVC, and Web Pages

ASP.NET offers three frameworks for creating web applications: ASP.NET Web Forms, ASP.NET MVC, and ASP.NET Web Pages. All three frameworks are stable and mature, and you can create great web applications with any of them.

Each framework targets a different audience or type of application. Which one you choose depends on a combination of your web development experience, what framework you're most comfortable with, and which is the best fit for the type of application you're creating. All three frameworks will be supported, updated, and improved in future releases of ASP.NET.

Here's an overview of each of the frameworks and some ideas for how to choose between them.

ASP.NET Web Forms (.aspx pages)

The Web Forms framework targets developers who prefer declarative and control-based programming, such as Microsoft Windows Forms (WinForms) and WPF/XAML/Silverlight. It offers a WYSIWYG designer-driven (drag-and-drop) development model, so it's popular with developers looking for a rapid application development (RAD) environment for web development. If you're new to web programming and are familiar with the traditional Microsoft RAD client development tools (for example, for Visual Basic and Visual C#), you can quickly build a web application without having expertise in HTML and JavaScript.

In particular, the Web Forms model provides the following features:

- An event model that exposes events which you can program like you would program a client application like WinForms or WPF.
- Server controls that render HTML for you and that you can customize by setting properties and styles.
- A rich assortment of controls for data access and data display.
- Automatic preservation of state (data) between HTTP requests, which makes it easy for a programmer who is accustomed to client applications to learn how to create applications for the stateless web.

Web Forms works well for small teams of Web developers and designers who want to take advantage of the large number of components available for rapid application development. In general, creating a Web Forms application requires less programming effort than creating the same application by using the ASP.NET MVC framework. The components (the Page class, controls, and so on) are tightly integrated and usually require less code than ASP.NET MVC applications. However, Web Forms is not just for rapid application development. There are many complex commercial apps and app frameworks built on top of Web Forms.

Because a Web Forms page and the controls on the page automatically generate much of the markup that's sent to the browser, you don't have the kind of fine-grained control over the HTML that the other ASP.NET models offer. An event-driven, control-focused model hides some of the behavior of HTML and HTTP. For example, it's not always possible to specify exactly what markup might be generated by a control.

The Web Forms model doesn't lend itself as readily as ASP.NET MVC to patterns-based development, separation of concerns, and automated unit testing. If you want to write code factored that way, you can; it's just not as automatic as it is in the ASP.NET MVC framework. The ASP.NET Web Forms MVP project shows an approach that facilitates separation of concerns and testability while maintaining the rapid development that Web Forms was built to deliver. As an example of this in action, Microsoft SharePoint is built using Web Forms MVP.

ASP.NET MVC

ASP.NET MVC targets developers who are interested in patterns and principles like test-driven development, separation of concerns, inversion of control (IoC), and dependency injection (DI). This framework encourages separating the business logic layer of a web application from its presentation layer.

By dividing the application into the model (M), views (V), and controllers (C), ASP.NET MVC can make it easier to manage complexity in larger applications. With ASP.NET MVC, you can have multiple teams working on a web site because the code for the business logic is separate from the code and markup for the presentation layer — developers can work on the business logic while designers work on the markup and JavaScript that is sent to the browser.

With ASP.NET MVC, you work more directly with HTML and HTTP than in Web Forms. Web Forms tends to hide some of that by mimicking the way you would program a WinForms or WPF application. For example, Web Forms can automatically preserve state between HTTP requests, but you have to code that explicitly in MVC. The MVC model enables you to take complete control over exactly what your application is doing and how it behaves in the web environment.

MVC was designed to be extensible, providing power developers the ability to customize the framework for their application needs. In addition, the ASP.NET MVC source code is available under an OSI license.

MVC 4 includes a feature that helps you develop HTTP services that reach a broad range of clients, including browsers and mobile devices. For more information, see Getting Started with ASP.NET Web API. MVC 4 also helps you develop single-page applications (SPAs) that use client-side JavaScript for responsive client interaction. For more information, see Single Page Application Overview.

For more information about ASP.NET MVC, see ASP.NET MVC.

ASP.NET Web Pages (.cshtml and .vbhtml files)

ASP.NET Web Pages targets developers who want a simple web development story, along the lines of PHP. In the Web Pages model, you create HTML pages and then add server-based code to the page in order to dynamically control how that markup is rendered. Web Pages is specifically designed to be a lightweight framework, and it's the easiest entry point into ASP.NET for people who know HTML but might not have broad programming experience — for example, students or hobbyists. It's also a good way for web developers who know PHP or similar frameworks to start using ASP.NET.

Like Web Forms, Web Pages is oriented toward rapid development. Web Pages provides components called *helpers* that you can add to pages and that let you use just a few lines of code to perform tasks that would either be tedious or complex. For example, there are helpers to display database data, add a Twitter feed, log in using Facebook, add maps to a page, and so on.

Web Pages provides a simpler approach than Web Forms. If you look at a .cshtml or .vbhtml file, you can generally think of the logic as executing top-to-bottom in the file, as you would with PHP, SHTML, etc. And because .cshtml and .vbhtml files are essentially HTML files that have additional ASP.NET code in them, they lend themselves easily to adding client-side functionality via JavaScript and jQuery.

For more information about ASP.NET Web Pages, see ASP.NET Web Pages on the ASP.NET web site.

General Notes

All three ASP.NET frameworks are based on the .NET Framework and share core functionality of .NET and of ASP.NET.

For example, all three frameworks offer a login security model based around a membership API, and all three share the same facilities for managing requests, handling sessions, and so on that are part of the core ASP.NET functionality.

In addition, the three frameworks are not entirely independent, and choosing one does not preclude also using another. For example, MVC views are often written as .cshtml or .vbhtml files (using "Razor" syntax), which means they can take advantage of some of the Web Pages features like helpers. Since the frameworks can also coexist in the same web application, it's not uncommon to see individual components of an application written using different frameworks. For example, the bulk of a site might be written in MVC, but the data access portion of the site might be written using Web Forms because it's such an easy framework in which to perform data access. In these cases, the developers choose the hybrid solution that plays to their strengths and makes their lives the easiest for their particular scenarios.

As of March 28, 2012, Microsoft has placed ASP.NET MVC 4, ASP.NET Web API, and ASP.NET Web Pages v2 (Razor syntax) under the open source Apache 2.0 license. ASP.NET Web Forms is not included. For more information, see ASP.NET MVC, Web API, Razor and Open Source (ScottGu's blog) and ASP.NET MVC 4, ASP.NET Web API and ASP.NET Web Pages v2 (Razor) now all open source with contributions (Scott Hanselman's blog).

The remaining sections of this topic provide an overview of ASP.NET features that are common to all three ASP.NET frameworks or unique to Web Forms.

Visual Web Developer

Visual Web Developer is a full-featured development environment for creating ASP.NET Web applications. Visual Web Developer offers you the following features:

- **Web page design** A powerful Web page editor that includes WYSIWYG editing and an HTML editing mode with IntelliSense and validation.
- **Page design features** Consistent site layout with master pages and consistent page appearance with themes and skins.
- **Code editing** A code editor that enables you to write code for your dynamic Web pages in Visual Basic or C#. The code editor includes syntax coloration and IntelliSense.
- **Testing and Debugging** A local Web server for testing and a debugger that helps you find errors in your programs.
- **Deployment** Tools to automate typical tasks for deploying a Web application to a hosting server or a hosting provider.

For more information, see Visual Studio Web Development Content Map.

Testing and Debugging

Visual Web Developer provides an ideal environment in which to build Web sites and then publish them to a hosting site. Using the development tools in Visual Web Developer, you can develop ASP.NET Web pages on your own computer. Visual Web Developer includes a local Web server that provides all the features you need to test and debug ASP.NET Web pages, without requiring Internet Information Services (IIS) to be installed.

When your site is ready, you can publish it to the host computer using the built-in Copy Web tool, which transfers your

files when you are ready to share them with others. Alternatively, you can precompile and deploy a Web site by using the **Build Web Site** command. The **Build Web Site** command runs the compiler over the entire Web site (not just the code files) and produces a Web site layout that you can deploy to a production server.

Note
The Build Web Site feature is not available in Visual Web Developer Express Edition.

Finally, you can take advantage of the built-in support for File Transfer Protocol (FTP). Using the FTP capabilities of Visual Web Developer, you can connect directly to the host computer and then create and edit files on the server.

ASP.NET Web Sites and ASP.NET Web Application Projects

Using Visual Studio, you can create different types of ASP.NET projects, which includes Web sites, Web applications, Web services, and AJAX server controls.

There is a difference between Web site projects and Web application projects. Some features work only with Web application projects, such as MVC and certain tools for automating Web deployment. Other features, such as Dynamic Data, work with both Web sites and Web application projects. For more information about the differences between Web application projects and Web site projects, see Web Application Projects versus Web Site Projects in Visual Studio.

ASP.NET API Reference

Some of the most important namespaces in the .NET Framework class library that pertain to ASP.NET are the following:

System.Web
: Provides classes and interfaces that enable browser-server communication. This namespace includes the HttpRequest class, which provides extensive information about the current HTTP request, the HttpResponse class, which manages HTTP output to the client, and the HttpServerUtility class, which provides access to server-side utilities and processes. System.Web also includes classes for cookie manipulation, file transfer, exception information, and output cache control.

System.Web.ApplicationServices
: Provides classes that provide access to ASP.NET forms authentication, roles, and profiles application services as Windows Communication Foundation (WCF) services.

System.Runtime.Caching
: Contains types that let you implement caching in .NET Framework applications.

System.Web.ClientServices
: Contains classes that support access to the ASP.NET login, roles, and profiles services from Windows-based applications.

System.Web.Configuration
: Contains classes that are used to programmatically manage ASP.NET configuration. (Most configuration settings

can be made in XML files.)

System.Web.DynamicData
: Contains classes that provide the core functionality for ASP.NET dynamic data and extensibility features that let you customize dynamic data behavior.

System.Web.Handlers
: Contains HTTP handler classes that process HTTP requests to a Web server. (An ASP.NET Web Forms page -- .aspx file -- is a special form of an HTTP handler.)

System.Web.Management
: Contains classes and interfaces for managing and monitoring the health of Web applications.

System.Web.Profile
: Contains classes that are used to implement the ASP.NET user profile in Web server applications.

System.Web.Query.Dynamic
: Contains classes that are used to parse expressions from a LinqDataSource control into a Language-Integrated Query (LINQ).

System.Web.RegularExpressions
: Provides regular expressions that are used to parse ASP.NET files. All members of the System.Web.RegularExpressions namespace are descendants of the Regex class. (You typically do not have to parse ASP.NET pages yourself.)

System.Web.Routing
: Provides classes that are used with URL routing, which enables you to use URLs that do not map to a physical file.

System.Web.Script
: Contains classes that provide client-script resource information.

System.Web.Script.Services
: Provides attributes to customize Web service support for using Ajax functionality in ASP.NET.

System.Web.Security
: Contains classes that are used to implement ASP.NET security in Web server applications.

System.Web.Services
: Consists of the classes that enable you to create XML Web services using ASP.NET and XML Web service clients. XML Web services are applications that provide the ability to exchange messages in a loosely coupled environment using standard protocols such as HTTP, XML, XSD, SOAP, and WSDL. XML Web services let you build modular applications that are interoperable across a broad variety of implementations, platforms, and devices.

System.Web.SessionState
: Contains classes and interfaces that enable storage of data specific to a single client during a single browser session on the server. Session state data is used to give the client the appearance of a persistent connection with the application.

System.Web.UI
: Provides classes and interfaces that enable you to create ASP.NET server controls and ASP.NET Web pages for the user interface of your ASP.NET Web applications. This namespace includes the Control class, which provides all HTML server controls, Web server controls, and user controls with a common set of functionality. It also includes the Page control, which is generated automatically whenever a request is made for an .aspx file in an ASP.NET Web application. Also included are classes which provide the server controls with data-binding functionality, the ability

to save the view state of a given control or page, and parsing functionality.

System.Web.UI.DataVisualization.Charting
: Contains types for the **Chart** Web server control.

System.Web.UI.Design.WebControls
: Contains classes that can be used to extend design-time support for Web server controls.

System.Web.UI.Design.WebControls.WebParts
: Contains classes that provide design-time support for controls derived from classes in the System.Web.UI.WebControls.WebParts namespace.

System.Web.UI.HtmlControls
: Contains a collection of classes that enable you to create HTML server controls on a Web Forms page. HTML server controls run on the server and map directly to standard HTML tags supported by most browsers. This enables you to programmatically control the HTML elements on a Web Forms page.

System.Web.UI.WebControls
: Contains classes that enable you to create Web server controls on a Web page. Web server controls run on the server and include form controls such as buttons and text boxes. They also include special-purpose controls such as a calendar. Because Web server controls run on the server, you can programmatically control these elements. Web server controls are more abstract than HTML server controls. Their object model does not necessarily reflect HTML syntax.

System.Web.UI.WebControls.WebParts
: Contains an integrated set of classes and interfaces for creating Web pages whose appearance and behavior can be modified (personalized) by end users. The user-defined settings for each page are saved for future browser sessions.

System.Web.Util
: Contains classes that enable callback methods to be run under the scope of a transaction and that enable work to be posted to separate threads.

ASP.NET MVC Reference
: This topic provides links to four namespaces that are used by the MVC framework.

For a complete list of .NET Framework namespaces, with links to API reference topics for them, see .NET Framework Class Library.

Page and Controls Framework

The ASP.NET Web Forms page and controls framework is a programming framework that runs on a Web server to dynamically produce and render ASP.NET Web pages. ASP.NET Web pages can be requested from any browser or client device, and ASP.NET renders markup (such as HTML) to the requesting browser. As a rule, you can use the same page for multiple browsers, because ASP.NET renders the appropriate markup for the browser making the request. However, you can design your ASP.NET Web page to target a specific browser and take advantage of the features of that browser.

ASP.NET Web Forms pages are completely object-oriented. Within ASP.NET Web forms pages you can work with HTML elements using properties, methods, and events. The ASP.NET page framework removes the implementation details of the separation of client and server inherent in Web-based applications by presenting a unified model for responding to client events in code that runs at the server. The framework also automatically maintains the state of a page and the controls on

that page during the page processing life cycle. For more information see ASP.NET Web Forms Pages Overview.

The ASP.NET page and controls framework also enables you to encapsulate common UI functionality in easy-to-use, reusable controls. Controls are written once, can be used in many pages, and are integrated into the ASP.NET Web page that they are placed in during rendering.

The ASP.NET page and controls framework also provides features to control the overall look and feel of your Web site via themes and skins. You can define themes and skins and then apply them at a page level or at a control level. For more information, see ASP.NET Themes and Skins.

In addition to themes, you can define master pages that you use to create a consistent layout for the pages in your application. A single master page defines the layout and standard behavior that you want for all the pages (or a group of pages) in your application. You can then create individual content pages that contain the page-specific content you want to display. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page. For more information see ASP.NET Master Pages.

The ASP.NET page framework also enables you to define the pattern for URLs that will be used in your site. This helps with search engine optimization (SEO) and makes URLs more user-friendly. For more information, see ASP.NET Routing.

The ASP.NET page and control framework is designed to generate HTML that conforms to accessibility guidelines. For more information, see Accessibility in Visual Studio and ASP.NET.

ASP.NET Compiler

All ASP.NET code is compiled, which enables strong typing, performance optimizations, and early binding, among other benefits. Once the code has been compiled, the common language runtime further compiles ASP.NET code to native code, providing improved performance.

ASP.NET includes a compiler that will compile all your application components including pages and controls into an assembly that the ASP.NET hosting environment can then use to service user requests. For more information, see ASP.NET Compilation Overview.

Security Infrastructure

In addition to the security features of .NET, ASP.NET provides an advanced security infrastructure for authenticating and authorizing user access as well as performing other security-related tasks. You can authenticate users using Windows authentication supplied by IIS, or you can manage authentication using your own user database using ASP.NET forms authentication and ASP.NET membership. Additionally, you can manage the authorization to the capabilities and information of your Web application using Windows groups or your own custom role database using ASP.NET roles. You can easily remove, add to, or replace these schemes depending upon the needs of your application. For more information, see the following topics:

- ASP.NET Security
- Managing Users by Using Membership
- Managing Authorization Using Roles
- Forms Authentication Provider

ASP.NET always runs with a particular Windows identity so you can secure your application using Windows capabilities such as NTFS Access Control Lists (ACLs), database permissions, and so on. For more information about the identity of ASP.NET, see Configuring ASP.NET Process Identity and ASP.NET Impersonation.

Web Forms State-Management Facilities

ASP.NET provides intrinsic state management functionality that enables you to store information between page requests, such as customer information or the contents of a shopping cart. You can save and manage application-specific, session-specific, page-specific, user-specific, and developer-defined information. This information can be independent of any controls on the page.

ASP.NET offers distributed state facilities, which enable you to manage state information across multiple instances of the same application on one computer or on several computers. For more information, see ASP.NET State Management Overview.

ASP.NET Configuration

ASP.NET applications use a configuration system that enables you to define configuration settings for your Web server, for a Web site, or for individual applications. You can make configuration settings at the time your ASP.NET applications are deployed and can add or revise configuration settings at any time with minimal effect on operational Web applications and servers. ASP.NET configuration settings are stored in XML-based files. Because these XML files are text files, it is simple to make configuration changes to your Web applications. You can extend the configuration scheme to suit your requirements. For more information, see ASP.NET Configuration Overview.

Health Monitoring and Performance Features

ASP.NET includes features that enable you to monitor health and performance of your ASP.NET application. ASP.NET health monitoring enables reporting of key events that provide information about the health of an application and about error conditions. These events show a combination of diagnostics and monitoring characteristics and offer a high degree of flexibility in terms of what is logged and how it is logged. For more information, see ASP.NET Health Monitoring Overview.

ASP.NET supports two groups of performance counters available to your applications:

- The ASP.NET system performance counter group
- The ASP.NET application performance counter group

For more information, see Monitoring ASP.NET Application Performance.

Debugging Support

ASP.NET takes advantage of the run-time debugging infrastructure to provide cross-language and cross-computer

debugging support. You can debug both managed and unmanaged objects, in addition to all languages supported by the common language runtime and script languages. For details, see ASP.NET Debugging.

In addition, the ASP.NET page framework provides a trace mode that enables you to insert instrumentation messages into your ASP.NET Web pages. For more information, see ASP.NET Tracing Overview.

Web Services Framework

ASP.NET supports Web services using Windows Communication Foundation. A Web service is a component that contains business functionality that enables applications to exchange information across firewalls by using standards like HTTP and XML messaging. Web services are not tied to a particular component technology or object-calling convention. As a result, programs written in any language, using any component model, and running on any operating system can access Web services. For more information, see Windows Communication Foundation Services and WCF Data Services in Visual Studio.

Extensible Hosting Environment and Application Life-Cycle Management

ASP.NET includes an extensible hosting environment that controls the life cycle of an application from when a user first accesses a resource (such as a page) in the application to the point at which the application is shut down. While ASP.NET relies on a Web server (IIS) as an application host, ASP.NET provides much of the hosting functionality itself. The architecture of ASP.NET enables you to respond to application events and create custom HTTP handlers and HTTP modules. For more information, see ASP.NET Application Life Cycle Overview for IIS 5.0 and 6.0.

Web Forms Extensible Designer Environment

ASP.NET includes enhanced support for creating designers for Web server controls for use with a visual design tool such as Visual Studio. Designers enable you to build a design-time user interface for a control, so that developers can configure your control's properties and content in the visual design tool. For more information, see ASP.NET Control Designers Overview.

ASP.NET Dynamic Data Scaffolding

ASP.NET Dynamic Data scaffolding is a framework built on ASP.NET Web Forms that lets you create data-driven ASP.NET Web applications easily. It does this by automatically discovering data-model metadata at run time and deriving UI behavior from it. A scaffolding framework provides a functional Web site for viewing and editing data. You can easily customize the scaffolding framework by changing elements or creating new ones to override the default behavior. Existing applications can easily integrate scaffolding elements together with ASP.NET pages.

You will find that you can get applications up and running more easily and with less code than before. At the same time, you can add custom features to ASP.NET Dynamic Data to accommodate your own requirements. For more information, see ASP.NET Dynamic Data.

See Also

Concepts

ASP.NET Application Life Cycle Overview for IIS 5.0 and 6.0

ASP.NET Web Forms Pages Overview

Other Resources

Visual Studio Web Development Content Map

Desktop Development

patterns & practices
proven practices for predictable results

Desktop development covers guidance for building the user experience portion of a solution. This includes guidance for building client-side applications, mobile device applications, Web applications, and rich internet applications (RIA). You will find guidance on building composite Windows Forms desktop and mobile applications, on building offline-capable applications, on application deployment, and on Web client application development.

Active Releases

- Prism 5.0. Learn how to create modular and maintainable WPF applications that are built to last and built for change using patterns such as MVVM and Event Aggregation. Includes 10 samples and library source and associated documentation.
- Developing a Windows Store business app using C#, XAML, and Prism for the Windows Runtime. This release provides guidance for devs who want to create business apps for the Windows 8 Store. It includes a reference implementation that demonstrates how to implement MVVM with navigation and app lifecycle management, manage application data, implement controls, validation, touch, search, tiles, tile notifications, and create accessible and localizable pages. It also provides guidance on testing your app and tuning its performance.
- Developing an end-to-end Windows Store app using C++ and XAML: Hilo (Windows). This guide discusses the design and implementation of Hilo, a photo app for Windows 8. Hilo teaches you how to use a modern coding style, asynchronous programming, and the Windows Runtime to build a world-ready app for the global market. It shows how to implement tiles, pages, controls, touch, navigation, file system queries, suspend/resume, and localization and how to use common architectural patterns such as Model-View-ViewModel (MVVM) and Model-View-Presenter (MVP). Hilo also demonstrates how to test your app and tune its performance.
- Developing an end-to-end Windows Store app using JavaScript: Hilo (Windows). The JavaScript version of the Hilo photo sample provides guidance to JavaScript developers who want to create a Windows 8 app using HTML5, CSS3, JavaScript, the Windows Runtime, and modern development patterns. Hilo comes with source code and documentation.
- Prism 4.1 for WPF and Silverlight. Prism provides best practice guidance for building Windows Presentation Foundation (WPF) and Silverlight, and Windows Phone client applications. Prism focuses on the patterns that support composite, extensible applications and test-driven development. It includes re-usable code and components, comprehensive documentation, QuickStarts, How-to topics, and a sample reference implementation. Version 4 of Prism was released in November 2010.
- Smart Client Architecture and Design Guide — June 2004

Please refer to the patterns & practices: Retired section for all retired offerings.

About Microsoft patterns & practices

Recommendations on how to design and develop custom applications using the Microsoft platform.

By using patterns & practices offerings, you can accelerate the design and development of your custom applications, reduce project technical risk, and position yourself to take advantage of future Microsoft technologies. Each patterns & practices offering contains a combination of written documentation and re-usable source code. Many also include a reference

implementation. This combination provides you with a solid starting point for your application, and a set of proven practices to aid your development.

For more information, visit the patterns & practices Developer Center.

Developing for Multiple Platforms with the .NET Framework

.NET Framework (current version)

You can develop apps for both Microsoft and non-Microsoft platforms by using the .NET Framework and Visual Studio.

Options for cross-platform development

To develop for multiple platforms, you can share source code or binaries, and you can make calls between .NET Framework code and Windows Runtime APIs.

If you want to...	Use...
Share source code between Windows Phone 8.1 and Windows 8.1 apps	**Shared projects** (Universal Apps template in Visual Studio 2013, Update 2). • Currently no Visual Basic support. • You can separate platform-specific code by using #**if** statements. For details, see: • Build apps that target Windows and Windows Phone by using Visual Studio (MSDN article) • Using Visual Studio to build Universal XAML Apps (blog post) • Using Visual Studio to Build XAML Converged Apps (video)
Share binaries between apps that target different platforms	**Portable Class Library projects** for code that is platform-agnostic. • This approach is typically used for code that implements business logic. • You can use Visual Basic or C#. • API support varies by platform. • Portable Class Library projects that target Windows 8.1 and Windows Phone 8.1 support Windows Runtime APIs and XAML. These features aren't available in older versions of the Portable Class Library. • If needed, you can abstract out platform-specific code by using interfaces or abstract classes. For details, see: • Cross-Platform Development with the Portable Class Library (MSDN article) • How to Make Portable Class Libraries Work for You (blog post) • Using Portable Class Library with Model-View-View Model (MSDN article)

	• App Resources for Libraries That Target Multiple Platforms (MSDN article) • .NET Portability Analyzer (Visual Studio extension)
Share source code between apps for platforms other than Windows 8.1 and Windows Phone 8.1	**Add as link** feature. • This approach is suitable for app logic that's common to both apps but not portable, for some reason. You can use this feature for C# or Visual Basic code. For example, Windows Phone 8 and Windows 8 share Windows Runtime APIs, but Portable Class Libraries do not support Windows Runtime for those platforms. You can use **Add as link** to share common Windows Runtime code between a Windows Phone 8 app and a Windows Store app that targets Windows 8. For details, see: • Share code with Add as Link (MSDN article) • How to: Add Existing Items to a Project (MSDN article)
Write Windows Store apps using the .NET Framework or call Windows Runtime APIs from .NET Framework code	**Windows Runtime APIs** from your .NET Framework C# or Visual Basic code, and use the .NET Framework to create Windows Store apps. You should be aware of API differences between the two platforms. However, there are classes to help you work with those differences. For details, see: • .NET Framework Support for Windows Store Apps and Windows Runtime (MSDN article) • Passing a URI to the Windows Runtime (MSDN article) • WindowsRuntimeStreamExtensions (MSDN API reference page) • WindowsRuntimeSystemExtensions (MSDN API reference page)
Build .NET Framework apps for non-Microsoft platforms	**Portable Class Library reference assemblies** in the .NET Framework, and a Visual Studio extension or third-party tool such as Xamarin. For details, see: • Portable Class Library now available on all platforms. (blog post) • Xamarin (Xamarin website)
Use JavaScript and HTML for cross-platform development	**Universal App templates** in Visual Studio 2013, Update 2 to develop against Windows Runtime APIs for Windows 8.1 and Windows Phone 8.1. Currently, you can't use JavaScript and HTML with .NET Framework APIs to develop cross-platform apps. For details, see:

	• JavaScript Project Templates • Porting a Windows Runtime app using JavaScript to Windows Phone

Solution Development Fundamentals

patterns & practices
proven practices for predictable results

Solution development fundamentals cover the cross-cutting aspects of solution development, such as security, caching, data access, validation, exception management, and so on. It also includes application architecture, development process, the software development life cycle (SDLC), and application life cycle guidance. You will find guidance and patterns that are generally applicable to solution development regardless of the specific architecture or scenario.

Active Releases

- Enterprise Library. Enterprise Library is a collection of application blocks that address common cross-cutting concerns that developers face when developing applications. The latest version of Enterprise Library (version 6) was released in April 2013 and includes two new application blocks (Semantic Logging Application Block and Transient Fault Handling Application Block) and many improvements.

- Semantic Logging 2.0. Semantic Logging can help to minimize the development effort required to implement structured event logging in your applications, and reduce the chances of inconsistency and errors when writing code that conforms to modern practice for generating logs containing semantically useful typed information. The Semantic Logging Application Block is a framework for capturing and manipulating events raised by applications, and storing the typed and structured information they contain in log files or other logging stores. Logs of this type make automated log parsing and monitoring much easier and more efficient.

- Unity. Unity is a lightweight and extensible dependency injection container with support for interception. We recently released Unity 3.0 (also part of the latest version of Enterprise Library) which added a number of key features, including registration by convention and support for Windows Store apps. Unity provides a mature and stable foundation for building high-quality, flexible, pattern-based libraries and reference implementations. It facilitates loosely-coupled design and help improve testability.

- Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence. This guide describes how to design and build applications and services that can take best advantage of SQL and NoSQL databases by combining them into a polyglot solution. It provides an end to end walkthrough of a business application that uses SQL Server in conjunction with a variety of NoSQL databases, showing how the designers selected the databases to closely match the various business requirements. This guide has an accompanying reference implementation of an online ordering system. The sample code illustrates how to implement a polyglot solution that stores data in a variety of SQL and NoSQL databases, and how to decouple the data access logic from the business logic of the application.

- Parallel Programming with Microsoft .NET This book describes patterns for parallel programming, with code examples, that use the new parallel programming support in the Microsoft® .NET Framework 4. This support is commonly referred to as the Parallel Extensions. You can use the patterns described in this book to improve your application's performance on multicore computers. Adopting the patterns in your code makes your application run faster today and also helps prepare for future hardware environments, which are expected to have an increasingly parallel computing architecture

- Parallel Programming with Microsoft Visual C++ This book describes patterns for parallel programming, with code examples, that use the parallel programming support in the Microsoft® Visual C++. The Parallel Patterns Library (PPL) and the Asynchronous Agents Library introduce a new programming model for parallelism that significantly simplifies the job. Behind the scenes are sophisticated algorithms that dynamically distribute computations on multicore architectures. In addition, Microsoft® Visual Studio® 2010 development system includes debugging and

analysis tools to support the new parallel programming model. You can use the patterns described in this book to improve your application's performance on multicore computers. Adopting the patterns in your code makes your application run faster today and also helps prepare for future hardware environments, which are expected to have an increasingly parallel computing architecture

- CQRS Journey. This guidance is designed to help you get started with the **Command & Query Responsibility Segregation** and the **Event Sourcing** patterns. The guide is a journal that describes the experiences of a development team with no prior CQRS proficiency in building, deploying (to Azure), and maintaining a sample real-world complex enterprise system to showcase various CQRS and ES concepts and techniques.

- A Guide to Claims–based Identity and Access Control, 2nd Edition. This guide gives you enough information to evaluate claims-based identity as a possible option when you're planning a new application or making changes to an existing one. It is intended for any architect, developer, or information technology (IT) professional who designs, builds, or operates Web applications and services that require identity information about their users.

- Building a Release Pipeline with Team Foundation Server 2012. This book shows how to use Team Foundation Server to build a release pipeline that is based on the Build – Deploy – Test – Release pattern. It uses an iterative approach that begins with a simple, largely unautomated pipeline and ends with a completely automated, continuous delivery pipeline. By continuous delivery, we mean that through techniques such as versioning, continuous integration, automation, and environment management, you will be able to decrease the time between when you first have an idea and when that idea is realized as software that's in production. Any software that has successfully gone through your release process will be software that is production ready, and you can give it to customers whenever your business demands dictate. There are also hands-on labs that accompany the book to show you how to build a release pipeline using Visual Studio ALM tools.

- Testing for Continuous Delivery with Visual Studio 2012. This guide provides an end-to-end walkthrough of the testing scenarios supported by the Visual Studio 2012 infrastructure. It will help testers and developers use Team Foundation Server effectively as an application lifecycle management solution for testing and supporting products.

- Microsoft Application Architecture Guide, 2nd Edition. This guide provides design-level guidance for the architecture and design of applications built on the .NET Framework.

- Performance Testing Guidance for Web Applications. This guide shows you an end-to-end approach for implementing performance testing for your Web applications.

Technical Articles

- Intercepting Asynchronous Methods Using Unity Interception
- Write Less Code and Play More Golf — Getting to Know Enterprise Library
- Inject Some Life into Your Applications — Getting to Know the Unity Application Block
- Getting to Know the Team System Management Model Designer
- Exploring the Factory Design Pattern
- Exploring the Observer Design Pattern

Please refer to the patterns & practices: Retired section for all retired offerings.

About Microsoft patterns & practices

Recommendations on how to design and develop custom applications using the Microsoft platform.

By using patterns & practices offerings, you can accelerate the design and development of your custom applications, reduce project technical risk, and position yourself to take advantage of future Microsoft technologies. Each patterns & practices offering contains a combination of written documentation and re-usable source code. Many also include a reference

implementation. This combination provides you with a solid starting point for your application, and a set of proven practices to aid your development.

For more information, visit the patterns & practices Projects.

Speech Technologies

This section of the MSDN Library provides resources to help you get started developing solutions that take advantage of Microsoft speech technologies.

Documentation

Whether you are creating applications for Windows, Kinect for Windows, Lync 2010, Windows Mobile, or the cloud, Microsoft has the right speech technology to add the power of speech to almost any user interface. Learn more about implementing Microsoft speech technologies by following the links below.

- For Windows and Windows Server 2008. Add speech to your Windows applications using managed-code and native-code APIs to manage the speech engines that are included in Windows and Windows Server 2008.
- Speech Platforms. Incorporate speech into applications that leverage Microsoft's redistributable Runtime and Runtime Languages (language packs that enable speech recognition or text-to-speech for a specific language).
- Embedded. Speech is an effective and natural way for people to interact with devices.
- Services. Develop speech-enabled applications that leverage real-time voice services in the cloud and free yourself from building, maintaining, and upgrading a voice services infrastructure.

Online Services

Download Visual Studio

Quick access

My samples

Upload a sample

Browse sample requests

Developer code samples

Download code samples and applications for Windows 8, Windows Phone, Microsoft Azure, Office, SharePoint, Silverlight and other products. You can also explore the Official Visual Studio C#, VB.NET, and 101 LINQ samples.

New samples are added frequently in JavaScript, C++, C#, Visual Basic, and F#.

Search for samples

Platform

- Desktop 5278
- Web 2808
- Cloud 2003
- Windows Store apps 1305
- Data 1249
- Phone 1055
- Xbox 121

Visual Studio version

- Visual Studio 2010 3222
- Visual Studio 2012 2308
- Visual Studio 2013 1505
- Visual Studio 2015 450
- Visual Studio 15 2

Programming language

- C# 6816
- VB.NET 2276
- C++ 1023
- JavaScript 769
- F# 62

Contributors

- Community 5526
- Microsoft 3915
- All-in-One Code Framework team 1632
- Windows platform team 823
- Office team 767
- Microsoft Azure team 55
- Visual Studio Platform team 39
- Windows Driver Kit team 2
- Windows Phone SDK 1

9441 results

Sort by: Popularity

LINQ - Query Execution

Featured

Visual Studio Product Team - Microsoft

This sample shows different uses of Query Execution

LINQ

C#

ASP.NET MVC Application Using Entity Framework Code First

(162)
Updated 10/22/2015
Released 4/12/2011
510,941 Downloads

Tom Dykstra - MSFT - Microsoft

A Visual Studio 2013 project which shows how to use the Entity Framework 6 in an ASP.NET MVC 5 web application project, using the Code First development approach.

ADO.NET Entity Framework, ASP.NET, ASP.NET MVC

C#

ASP.NET MVC Application with a Custom Layout

(18)
Updated 2/10/2016
Released 2/2/2016
83,244 Downloads

Bartosz Rachwał

A Visual Studio 2015 project which shows how use create an Asp.Net MVC Application with a custom layout. The AdminLTE layout (MIT License) from https://almsaeedstudio.com/preview has been selected for the customization.

C#, CSS, HTML5/JavaScript

C#

Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013 - Wingtip Toys

(83)
Updated 4/20/2016
Released 4/7/2012
316,527 Downloads

ErikReitan

This download has been created for the ASP.NET 4.5 Web Forms and Visual Studio 2013 (Wingtip Toys) tutorial series. This series of tutorials guides you through the steps required to create an ASP.NET Web Forms application using Visual Studio Express 2013 for Web and ASP.NET 4.5.

ASP.NET Web Forms, ASP.NET 4.5, Visual Studio Express 2013 for Web

C#

101 LINQ Samples

(226)
Updated 4/26/2012
Released 8/12/2011
295,576 Downloads

Visual Studio Product Team - Microsoft

Learn how to use LINQ in your applications with these code samples, covering the entire range of LINQ functionality and demonstrating LINQ with SQL, DataSets, and XML.

LINQ

C#

Simple Calculator

(58)
Updated 8/15/2012
Released 8/14/2012
226,841 Downloads

Houssem Dellai

What I build is just a simple calculator that will show beginners how much easy it is to develop applications using the .NET Framework. Through this sample, they will see some most useful algorithmic operations.

C#, WPF, XAML

C#

Getting Started with ASP.NET Web API (Tutorial Sample)

(37)
Updated 10/22/2013
Released 6/1/2012
143,038 Downloads

Mike.Wasson - Microsoft

Sample code for the following tutorial on asp.net:Getting Started with ASP.NET Web APIExcerpt from the tutorial:
HTTP is not just for serving up web pages. It is also a powerful platform for building APIs that expose services and data. HTTP is simple, flexible, and ubiquitous.

ASP.NET Web API

C#

Technology

C#	1369
ASP.NET	868
Windows Forms	752
WPF	625
Microsoft Azure	592
.NET Framework	578
XAML	498
Win32	449
Windows Phone 8	417
Silverlight	355

Topic

C#	705
Controls	493
User Interface	378
Data Access	240
Data Binding	239
Windows Forms	227
How to	203
ASP.NET	189
Microsoft Azure	187
Windows Store app	176

Samples Environments for Microsoft Chart Controls

Cephas Lin - Microsoft

The samples environments for Microsoft Chart Controls for .NET Framework 4 contain over 200 samples for both ASP.NET and Windows Forms, covering every major feature in Chart Controls for .NET Framework 4. See every major feature in action and learn at the same time.

ASP.NET, Windows Forms, .NET Framework 4

(25)
Updated 5/13/2014
Released 5/13/2014
139,093 Downloads
C#, HTML

Example of math expressions, variables, and if statements

Jakeinc

Have you just installed Visual Studio and are ready to begin your first code? Then this sample is for you. This sample explains string variables and integer variables, if else statements, and mathematical expressions. These basic skills are necessary in C/F#/C++/VB programming.

C#, C++, F#

(6)
Updated 2/8/2015
Released 4/22/2013
69,087 Downloads
C#, VB.NET, C++, F#

Hyper-V Remote Management Configuration Utility (HVRemote)

John Howard -MSFT

HVRemote reduces the manual configuration steps needed for Hyper-V Remote Management down to a few simple commands, and can diagnose common configuration errors.

Hyper-V

(105)
Updated 8/18/2015
Released 3/29/2013
199,992 Downloads

Official Visual Studio 2010 Samples for C# 4.0

Visual Studio Product Team - Microsoft

These are the official samples for Visual Studio 2010 C# 4.0.

COM, Office, LINQ

(103)
Updated 7/17/2012
Released 11/29/2011
561,122 Downloads
C#, VB.NET, C++

Couldn't find the sample you were looking for?
Click here to search and vote for a sample request.

1 - 10 of 9441 Items « First < Prev 1 2 3 4 5 6 7 8 9 10 Next > Last »

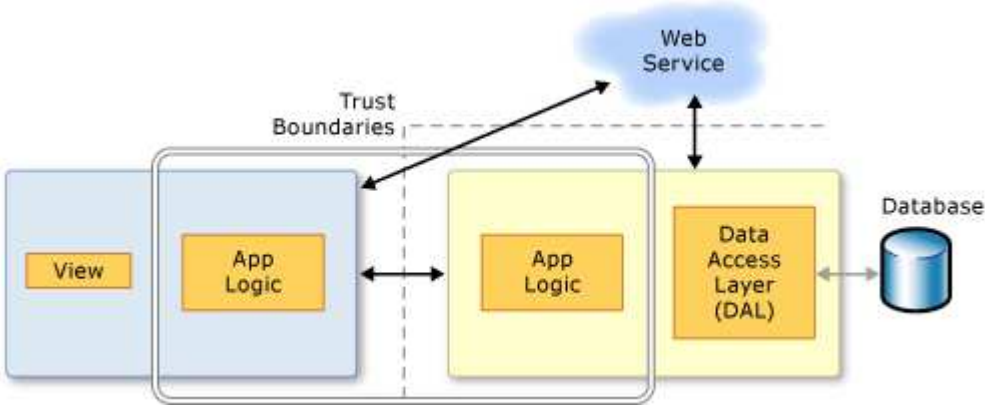
WCF RIA Services

WCF RIA Services

[WCF RIA Services Version 1 Service Pack 2 is compatible with either .NET framework 4 or .NET Framework 4.5, and with either Silverlight 4 or Silverlight 5.]

WCF RIA Services simplifies the development of n-tier solutions for Rich Internet Applications (RIA), such as Silverlight applications. A common problem when developing an n-tier RIA solution is coordinating application logic between the middle tier and the presentation tier. To create the best user experience, you want your RIA Services client to be aware of the application logic that resides on the server, but you do not want to develop and maintain the application logic on both the presentation tier and the middle tier. RIA Services solves this problem by providing framework components, tools, and services that make the application logic on the server available to the RIA Services client without requiring you to manually duplicate that programming logic. You can create a RIA Services client that is aware of business rules and know that the client is automatically updated with latest middle tier logic every time that the solution is re-compiled.

The following illustration shows a simplified version of an n-tier application. RIA Services focuses on the box between the presentation tier and the data access layer (DAL) to facilitate n-tier development with a RIA Services client.



RIA Services adds tools to Visual Studio 2010 that enable linking client and server projects in a single solution and generating code for the client project from the middle-tier code. The framework components support prescriptive patterns for writing application logic so that it can be reused on the presentation tier. Services for common scenarios, such as authentication and user settings management, are provided to reduce development time.

WCF Integration

In RIA Services, you expose data from the server project to client project by adding domain services. The RIA Services framework implements each domain service as a Windows Communication Foundation (WCF) service. Therefore, you can apply the concepts you know from WCF services to domain services when customizing the configuration. For more information, see Domain Services.

Securing a RIA Services Solution

To ensure that your application addresses the security concerns associated with exposing a domain service, you must carefully consider how you implement the domain service. For more information, see Building Secure Applications with

WCF RIA Services.

Tools and Documentation

The WCF RIA Services documentation require several prerequisite programs, such as Visual Studio 2010 and the Silverlight Developer Runtime and SDK, be installed and configured properly, in addition to WCF RIA Services and the WCF RIA Services Toolkit to work thorough the walkthroughs and how-to topics. They also require installing and configuring SQL Server 2008 R2 Express with Advanced Services and installing the AdventureWorks OLTP and LT database.

Detailed instructions for the satisfaction of each of these prerequisites are provided by the topics within the Prerequisites for WCF RIA Services node. Follow the instructions provided there before proceeding with this walkthrough to ensure that you encounter as few problems as possible when working through this RIA Services walkthroughs.

Topics

Prerequisites for WCF RIA Services

- Walkthrough: Installing and Configuring SQL Server 2008 R2 Express with Advanced Services
- Walkthrough: Installing the AdventureWorks OLTP and LT sample databases

Creating RIA Services Solutions

- Walkthrough: Taking a Tour of RIA Services
- Walkthrough: Creating a RIA Services Solution
- Walkthrough: Creating a RIA Service with the Code First Approach
- Walkthrough: Using the Silverlight Business Application Template
- Walkthrough: Creating a RIA Services Class Library
- Walkthrough: Localizing a Business Application
- How to: Create a Domain Service that uses POCO-defined Entities
- How to: Add or Remove a RIA Services Link
- Using the Domain Service Wizard

Building Secure Applications with WCF RIA Services

Deploying and Localizing a RIA Services Solutions

- Troubleshooting the Deployment of a RIA Services Solution

- Troubleshooting the Deployment of a RIA Services Solution
- Walkthrough: Localizing a Business Application

Middle Tier

- Domain Services
  - Walkthrough: Adding Query Methods
  - How to: Add Business Logic to the Domain Service
  - How to: Create a Domain Service that uses POCO-defined Entities
  - How to: Use HTTPS with a Domain Service
- Data
  - Compositional Hierarchies
  - Presentation Models
  - Inheritance in Data Models
  - Complex Types
  - Shared Entities
  - Walkthrough: Sharing Entities between Multiple Domain Services
  - How to: Add Metadata Classes
  - How to: Validate Data
  - Managing Data Concurrency
  - 1. How to: Enable Optimistic Concurrency Checks
    2. How to: Add Explicit Transactions to a Domain Service
- Shared Code
  - How to: Share Code through Source Files
  - Walkthrough: Creating a RIA Services Class Library

Silverlight Clients

- Client Code Generation
- DomainContext and Operations
- DomainDataSource

- Error Handling on the Client
- Customizing Generated Code
  - How to: Add Computed Properties on the Client

Accessing non-Silverlight Clients

- ASP.NET Clients
- Walkthrough: Using the Domain Service in ASP.NET Applications

Authentication, Roles, and Profiles

- How to: Enable Authentication in RIA Services
- How to: Enable Roles in RIA Services
- How to: Enable Profiles in RIA Services
- How to: Create a Custom Authorization Attribute
- Walkthrough: Using Authentication Service with Silverlight Business Application
- Walkthrough: Using Authentication Service with Silverlight Navigation Application

End-to-EndScenarios

- Walkthrough: Retrieving and Displaying Data From a Domain Service
- Walkthrough: Editing Data From a Domain Service
- Walkthrough: Displaying Data in a Silverlight Business Application
- Walkthrough: Displaying Related Data in a Silverlight Business Application

Reference

See Also

Other Resources

Offline RIA Services documentation

What's New in Visual Studio 2015

Visual Studio 2015

Welcome to Visual Studio 2015, an integrated suite of developer productivity tools, cloud services, and extensions that enable you and your team to create great apps and games for the web, for Windows Store, for the desktop, for Android, and for iOS.

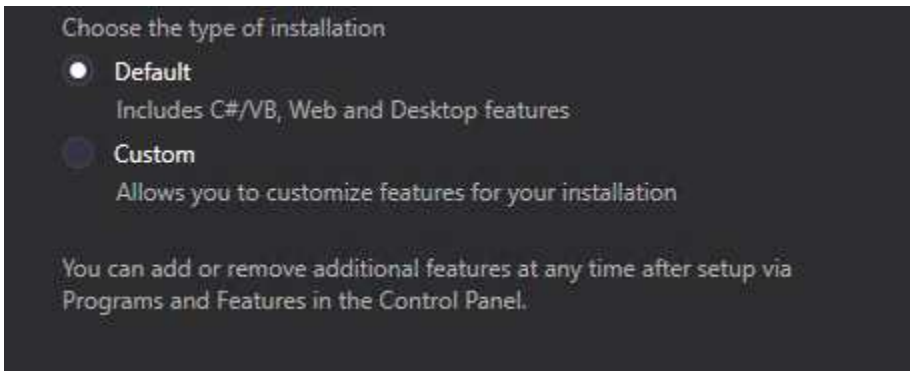
This page highlights some of the most important features that are new since Visual Studio 2013 RTM, including features that were first introduced in one of the Visual Studio 2013 updates. For a complete list of what's new in Visual Studio 2015, see the Release Notes.

To find out more about the many improvements and new features in Visual Studio ALM, see What's new for Application Lifecycle Management in Visual Studio 2015.

A new setup experience

Download Visual Studio Community or compare Visual Studio editions

The Visual Studio 2015 setup experience has been componentized so that you only have to install the parts that you need. This makes installation faster for many common scenarios involving .NET or Web development. If you do other types of development, such as cross-platform mobile development, or you work in C++ or F#, choose **Custom** installation and then choose the components and optional third-party SDKs that you require. You can also install any of the custom components later. For example, if you choose Basic installation, and then attempt to create a new C++ project, you will be prompted to download the C++ development tools.

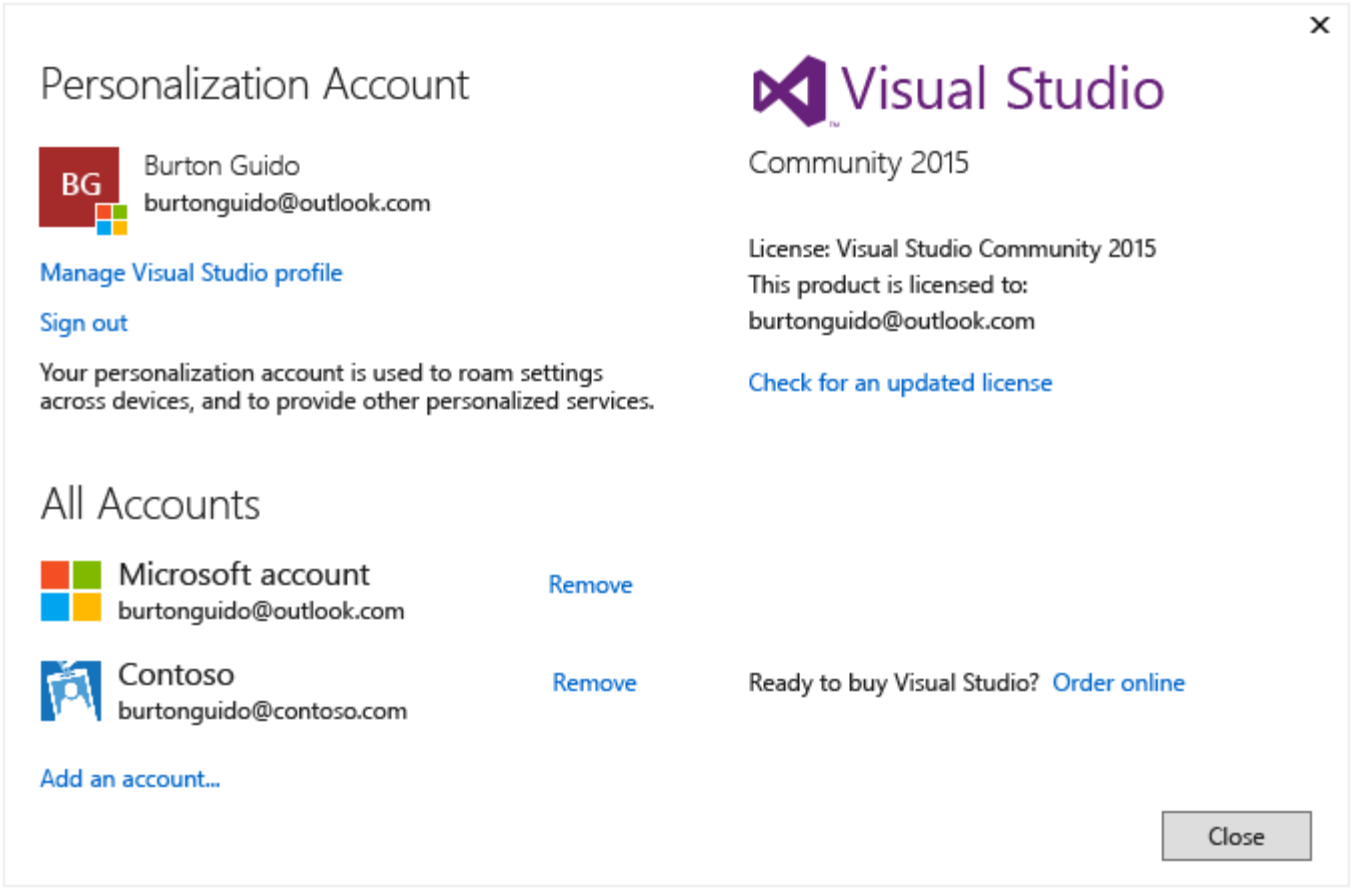


Sign in across multiple accounts

With Visual Studio 2015, the new streamlined sign-in experience is designed to greatly simplify your access to online resources, even when you have multiple Visual Studio accounts. After you sign-in to Visual Studio, you are automatically signed in to all instances of Visual Studio 2015 and Blend on your machine. Signing in automatically starts roaming your settings for you. In Visual Studio 2015, your account is shared across features so, as long as you have a good token, you can access your Visual Studio Team Services account(s) from **Team Explorer**, and resources and websites from your

Microsoft Azure subscription in Server Explorer. You'll also see your Azure resources in the New Project Dialog for Application Insights projects, and you'll see your Azure Mobile, Azure Storage, Microsoft Office 365 and Saleforce.com developer accounts in the new **Add a Connected Service** dialog.

You can work with multiple user accounts in Visual Studio by adding them as you go or through the new Account Manager. Then, you can switch between those accounts on the fly when connecting to services or accessing online resources. Visual Studio remembers the accounts you add so you can use them from any instance of Visual Studio or Blend. Visual Studio will also roam the list of accounts (though we won't roam your valuable credentials) with your Personalization account so you can quickly start working with one of those accounts on another device. Of course, you can remove accounts from the Account Settings dialog at any time. To get started, see Work with multiple user accounts.



Choose your target platform(s)

Visual Studio 2015 supports cross-platform mobile device development. You can write apps and games that target iOS, Android, and Windows and share a common code base, all from within the Visual Studio IDE. You'll see all these new project types in the File, New Project dialog.

And—of course—support for classic desktop applications is better than ever, with lots of improvements to languages, libraries, and tools.

Cross-platform mobile apps in C# with Xamarin for Visual Studio

Xamarin is a mobile framework that enables you to write code in C# that binds natively to iOS and Android APIs. Microsoft has partnered closely with Xamarin on their release of Xamarin for Visual Studio, an extension that enables you to develop for Android, iOS, and Windows Phone in a single solution with shared code. With Xamarin, you'll use

one language and one code base with minimal deltas between the platforms. Xamarin for Visual Studio is supported on Visual Studio 2010 and later. The starter edition of Xamarin is is included in Visual Studio 2015. To get started, see Build cross-platform apps with Xamarin in Visual Studio.

Cross-platform mobile apps in HTML/JavaScript with Apache Cordova

Visual Studio Tools for Apache Cordova is the result of close collaboration between Microsoft and the open source Apache Cordova community. The tools enable cross-platform mobile development using HTML, CSS, and JavaScript (or Typescript). You can target Android, iOS, and Windows with a single code base and enjoy the richness of the Visual Studio IDE including JavaScript IntelliSense, the DOM Explorer, JavaScript Console, breakpoints, watches, locals, Just My Code, and more. With Visual Studio Tools for Apache Cordova, your apps have access to native device capabilities on all platforms through plugins that provide a common JavaScript API. To get started, see Get Started with Visual Studio Tools for Apache Cordova.

Cross-platform mobile games in C# with Unity

Unity is a widely-used platform for multiplatform 2D and 3D game development. You can write your game in C# and run it natively on Android, iOS, Windows Phone, and many other platforms. Visual Studio Tools for Unity is an extension that integrates Unity with the Visual Studio IDE. With this extension, you get all the features of the Visual Studio IDE and debugger, in addition to productivity features that are designed for Unity developers. Visual Studio Tools for Unity 2.0 Preview 2 adds support for Visual Studio 2015, in addition to a number of new features, such as better visualization for objects in the Locals and Watch windows. Microsoft has recently acquired SyntaxTree, the creators of Visual Studio Tools for Unity. To download Visual Studio Tools for Unity 2.0 Preview 2, and for more information about Visual Studio Tools for Unity, see Visual Studio Tools for Unity 2.0.

Cross-platform apps and libraries for native C++

C++ is a language available natively across most mobile devices. You can use it to write cross-platform shared code libraries that can be built for multiple mobile platform targets. You can even create whole mobile apps in C++. Visual C++ gives you the tools to edit, build, deploy, and debug your cross-platform code. In addition to templates for Windows apps, you can create projects from templates for Android Native Activity apps, iOS apps, or shared code library projects for multiple platforms that include Xamarin hybrid apps. Platform-specific IntelliSense enables you to explore APIs and generate correct code for Android, iOS, or Windows targets. You can configure your build for x86 or ARM native platforms, and deploy your code to an iOS simulator or to iOS devices on a network-attached Mac, to directly attached Android devices, or use the performant Microsoft Visual Studio Emulator for Android for testing. You can set breakpoints, watch variables, view the stack and step through C++ code in the Visual Studio debugger. You can share all except the most platform-specific code across multiple app platforms, and build for them all with one solution in Visual Studio.

To get started on cross-platform C++, see Build cross-platform apps with Visual C++

Universal Windows apps for any Windows 10 device

With the Universal Windows Platform and our one Windows core, you can run the same app on any Windows 10 device from phones to desktops. Create these Universal Windows apps with Visual Studio 2015 and the Universal Windows App Development tools.

Run your app on a Windows 10 phone, a Windows 10 desktop, or an Xbox. It's the same app package! With the introduction of the Windows 10 single, unified core, one app package can run across all platforms. Several platforms have Extension SDKs that you can add to your app to take advantage of platform specific behaviors. For example, an extension SDK for mobile handles the back button being pressed on a Windows phone. If you reference an Extension SDK in your project, then just add runtime checks to test if that SDK is available on that platform. That's how you can have the same app package for each platform!

Use C#, Visual Basic, C++ or JavaScript to create these Universal Windows apps.

Web

ASP.NET 5 is a major update to MVC, WebAPI and SignalR, and runs on Windows, Mac, and Linux. ASP.NET 5 has been designed from the ground up to provide you with a lean and composable .NET stack for building modern cloud-based apps. The Visual Studio 2015 tooling is more closely integrated with popular web development tools such as Bower and Grunt. To get started, see the many blog posts on the NET Web Development and Tools Blog.

Classic desktop and Windows Store

Visual Studio 2015 continues to support classic desktop and Windows Store development. As Windows evolves, Visual Studio will evolve along with it. In Visual Studio 2015, the libraries and languages for .NET as well as C++ have made significant advances that are applicable to all versions of Windows.

The .NET Framework

The Microsoft .NET Framework 4.6 offers about 150 new APIs and 50 updated APIs to enable more scenarios. For example, more collections now implement IReadOnlyCollection(Of T) making them easier to use. In addition, ASP.NET 5, mentioned previously, offers a lean .NET platform for building modern cloud-based apps.

Windows Store apps written in C# that target the .NET Framework can now take advantage of .NET Native, which compiles apps to native code rather than IL, and .NET Framework 4.6 also adds RyuJIT, a 64-bit Just-In-Time (JIT) compiler.

The new C# and VB compilers ("Roslyn") significantly speed up compile times and provide comprehensive code

analysis APIs. Visual Studio 2015 takes advantage of Roslyn to provide more refactorings including inline rename, analyzers, and quick fixes.

The C# and Visual Basic languages both contain many smallish improvements in the core language and in IDE support. These improvements all add up to make your .NET coding experience even more intuitive, convenient, and productive.

For more information, see What's New in the .NET Framework and the .NET Blog.

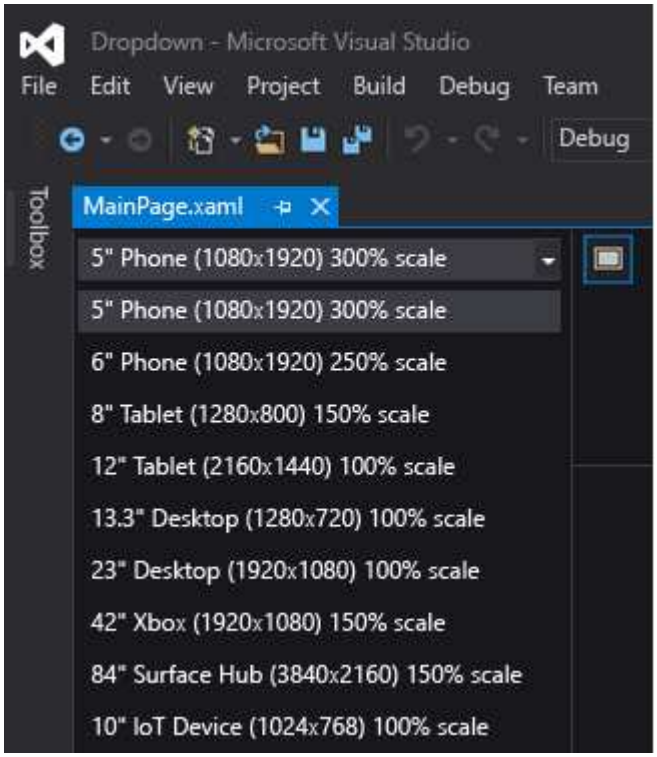
C++

Visual C++ provides significant advances in C++11/14 language conformance, support for cross-platform mobile device development, support for resumable functions and await (currently planned for standardization in C++17), improvements and bug fixes in the C Runtime Library (CRT) and C++ standard library (STL) implementations, resizeable dialogs in MFC, new compiler optimizations, better build performance, new diagnostics capabilities and new productivity tools in the code editor.

For more information, see What's New for Visual C++ in Visual Studio 2015 and the Visual C++ Blog.

Device Preview menu bar

In Universal Windows Platform projects, the device preview menu bar enables you to see how your XAML-based UI will render on various screen sizes.



Visual Studio Graphics Diagnostics

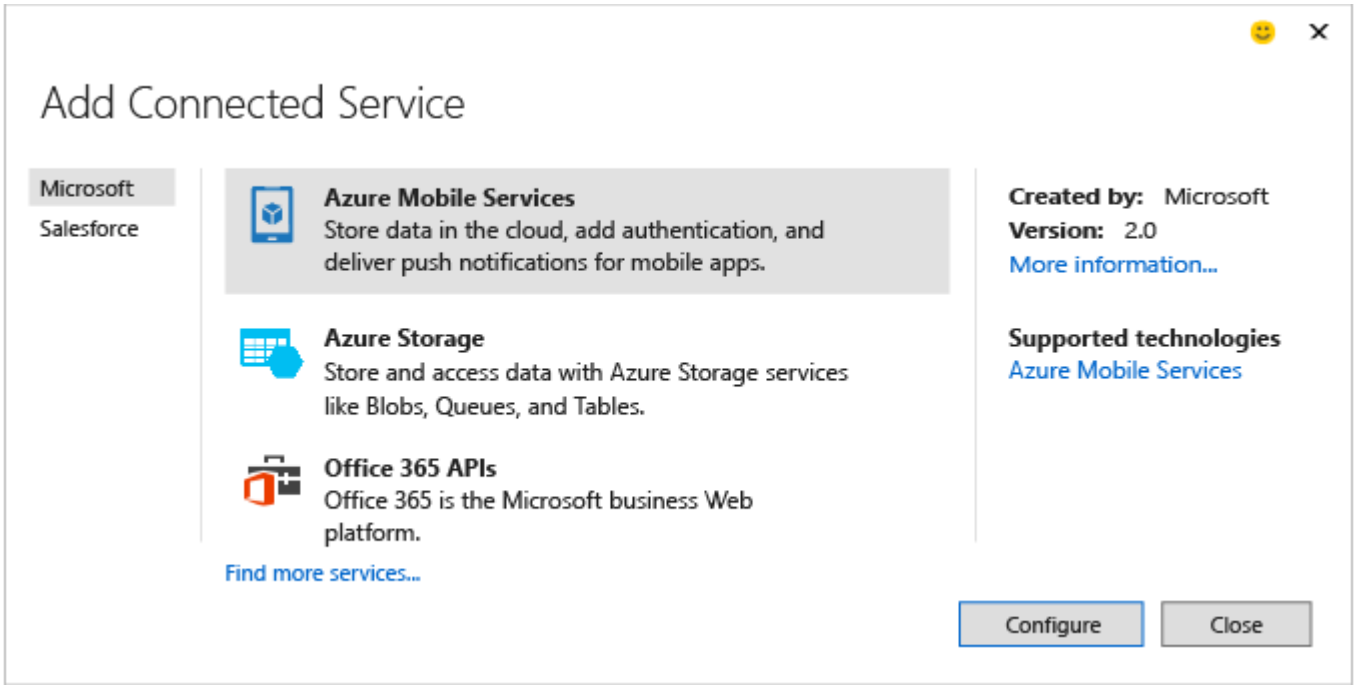
Since Visual Studio 2013, Visual Studio Graphics Diagnostics has added many new features, including Frame Analysis, Windows Phone support, shader edit & apply, and command line capture tools. It has also added support for debugging DirectX12 apps. For more information, see Visual Studio Graphics Diagnostics.

Connect to Services

Visual Studio 2015 makes it easier than ever to connect your app to services. The new Add Connected Service wizard configures your project, adds the necessary authentication support, and downloads the necessary NuGet packages to get you started coding against your service quickly and painlessly. The Add Connected Service wizard also integrates with the new Account Manager to make it easy to work with multiple user accounts and subscriptions. In Visual Studio 2015, support for the following services is provided out of the box (assuming that you have an account):

1. Azure Mobile Services
2. Azure Storage
3. Office 365 (mail, contacts, calendars, files, users & groups)
4. Salesforce

New services will be added on an ongoing basis, and you can discover those by clicking the "Find new services link" in the wizard.



Design your UI

The Blend experience for designing XAML user interfaces has been significantly enhanced. Blend has been completely

redesigned to provide a more intuitive UI, more powerful XAML editing capabilities including IntelliSense, and better integration with Visual Studio. For more information, see Designing XAML in Visual Studio.

Cross-platform debugging support

You can use Visual Studio to create and debug native mobile apps that run on Windows, iOS, and Android devices. Use the Visual Studio Emulator for Android, or connect a device and debug your code directly in Visual Studio.

- **JavaScript / Cordova**. Use the Visual Studio Tools for Apache Cordova to build native apps for Windows, iOS, and Android with JavaScript.

  Debug Your App Built with Visual Studio Tools for Apache Cordova in the MSDN Library is a detailed look at Visual Studio debugging support for Cordova.

- **C# / Xamarin**. Use Xamarin to build native apps for Windows, iOS, and Android in Visual Studio with C#.

  Debugging (iOS) and Debug on Device in the Xamarin developer guides describe the debugging experience.

- **C++ / Android**. Use the Visual C++ for Cross-Platform Mobile Development templates along with third-party tools like the Android NDK to create native apps for Windows and Android.

Debugging and Diagnostics

For information about what's new in debugging, see What's New for the Debugger in Visual Studio 2015.

For information about what's new in diagnostics, see What's New in Diagnostic Tools.

The following are new or improved tools that perform different types of diagnosis and analysis on your code:

PerfTips

PerfTips display the execution time of methods during debugging, enabling you to quickly spot bottlenecks without having to invoke the profiler. To get started, see PerfTips: Performance Information at-a-glance while Debugging with Visual Studio

Error List

The error list now supports filtering on any column. It also shows a live view of errors, warnings, and code analysis across your entire C# or Visual Basic solution as you type, even when a code change produces thousands of warnings. The new Error List is back-compatible with existing usage. For more information, see Error List Window.

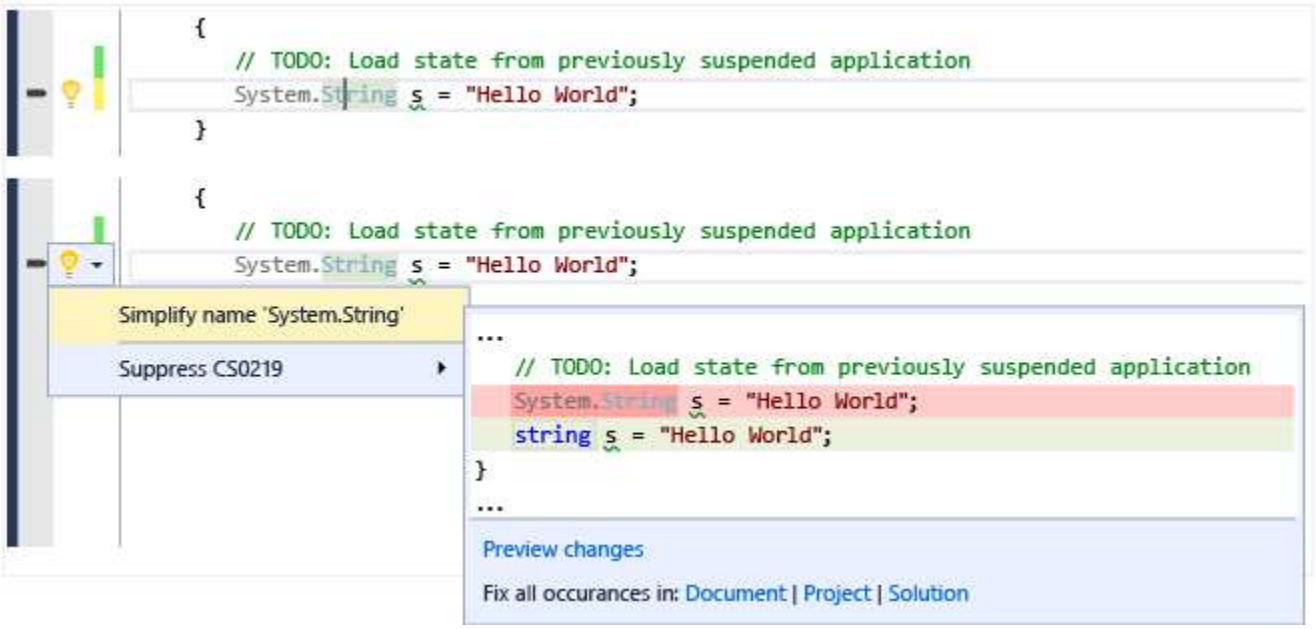
GPU Usage Tool

The GPU Usage Tool helps you collect and analyze GPU usage data in DirectX apps and games and troubleshoot

whether performance bottlenecks are originating in the CPU or GPU. To get started with the tool, see the Visual C++ team blog post.

Live code analysis (light bulbs)

The new Roslyn compiler for C# and Visual Basic not only provides faster compile times—it also enables completely new scenarios such as live code analysis, which provide rich and customizable feedback and suggestions directly inside the code editor as you type. In Visual Studio 2015, light bulbs display in the left margin (when using the keyboard) or a tool tip (when hovering over an error with the mouse). The light bulb tells in real time that the compiler (possibly using a custom rule set) has detected an issue in your code and also has a suggestion for how to fix the issue. When you see a light bulb, click on it for actionable suggestions.



Enjoy these additional IDE improvements

Synchronized Settings (Roaming Settings)

Visual Studio 2013 introduced Synchronized Settings for some of the most commonly configured settings such as Text Editor, Keybindings, Theme & Fonts & Colors, Startup, and Environment Aliases. Visual Studio 2015 improves on this experience by synchronizing more of your settings and synchronizing settings across Visual Studio family of applications like Professional, Enterprise, Express SKUs, and Blend. When you sign into Visual Studio 2015 for the first time with the same account as you used in Visual Studio 2013, you will see your synchronized settings applied from Visual Studio 2013. You can access your settings by typing "sync" in **Quick Launch**, or navigating to **Tools > Options > Environment > Synchronized Settings**.

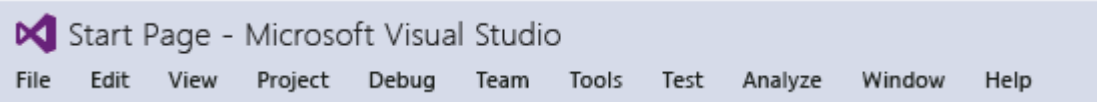
Automatic Extension Updates

Your installed Visual Studio extensions will now be automatically updated when a new version is available on the Visual

Studio Gallery. See Finding and Using Visual Studio Extensions for details on how you can customize automatic extension updates.

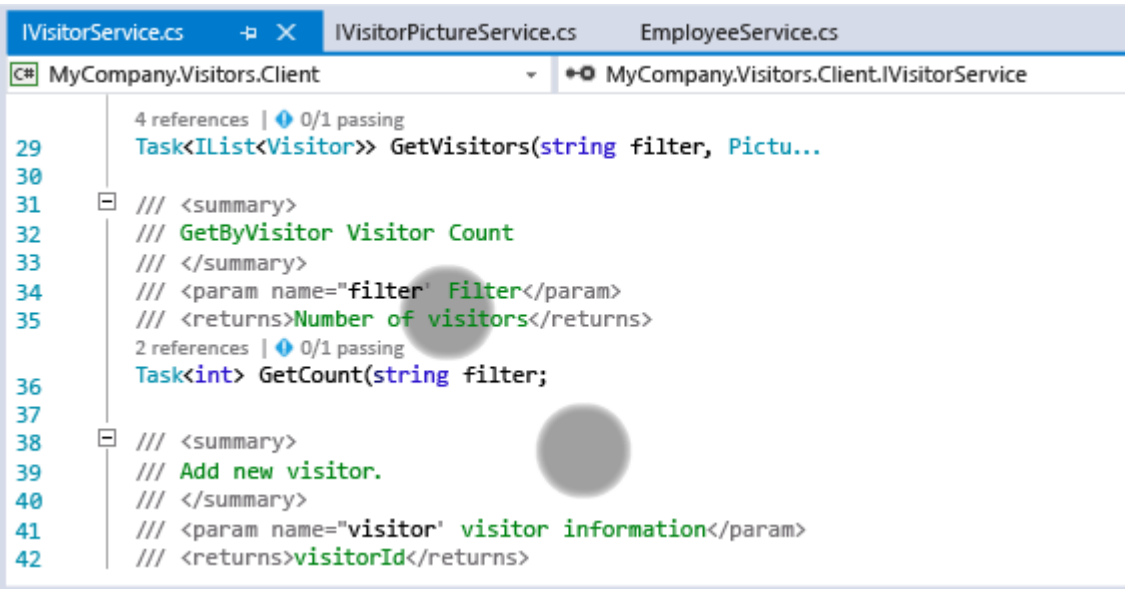
Title Case Menus

You spoke, we listened. Visual Studio menus are once again title-case by default. However if you happen to like the ALL CAPS style, you can set it on start up or in the **Tools > Options > General** property page:



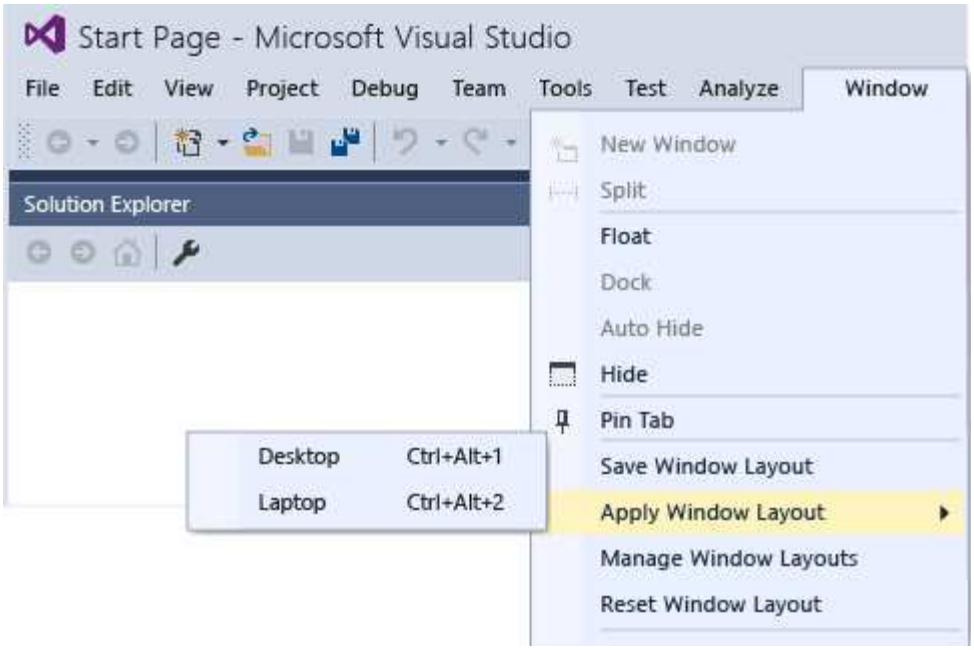
High Resolution Images and Touch Support

The Visual Studio IDE now has true high resolution images on denser displays (in areas like menus, context menus, tool window command bars, and in some projects in Solution Explorer). And on a touch-screen in the Visual Studio code editor window, you can now use gestures such as touch and hold, pinch, tap and so on to zoom, scroll, select text, and invoke context menus.



Custom Layouts

You can create store and roam custom window layouts. For example, you can define one preferred layout for use on your desktop machine, and different layout for use on a laptop or small screen device. Or you may prefer one layout for a UI project and another for a database project. Key bindings enable you to rapidly switch between layouts. These layouts are available on any instance of Visual Studio when you are signed in. For more information, see Create custom window layouts.



Notification Hub

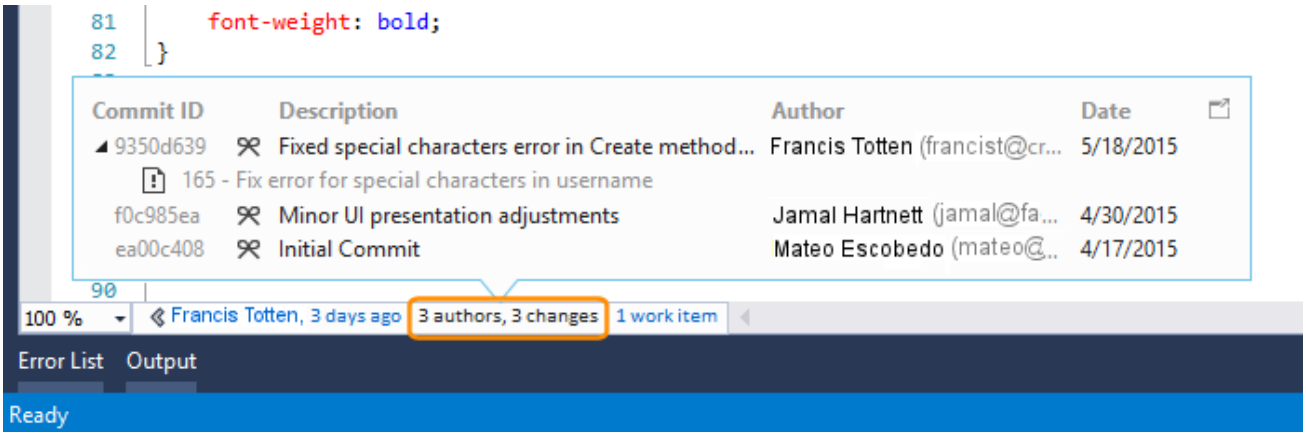
The UI for the notification hub has been streamlined to make it easier to scan quickly. Additional kinds of notifications have been added including performance issues, rendering issues, and crashes, and you can now tell Visual Studio to stop showing a notification. For more information, see Visual Studio Notifications.

CodeLens: Find what happened to your code (Enterprise and Professional editions only)

Stay focused on your work while you find information about your code - without leaving the editor. You can review changes and other history for work items, bugs, code reviews, and so on for code that's stored in Visual Studio Team Services (VSTS) or in Team Foundation Server (TFS).

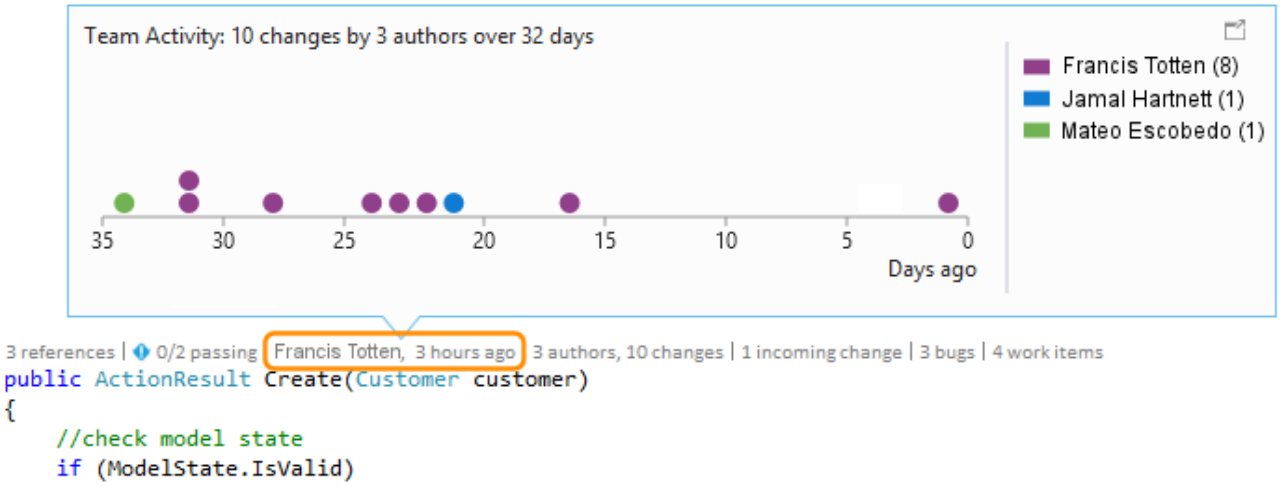
In Visual Studio Enterprise and Visual Studio Professional, you can now:

- Get history for an entire code file in the Visual Studio editor.



- See a graph that shows the people who changed your code. This can help you find patterns in your team's

changes and assess their impact.



- Easily see when your code was last changed.
- Find changes in other branches that affect your code.

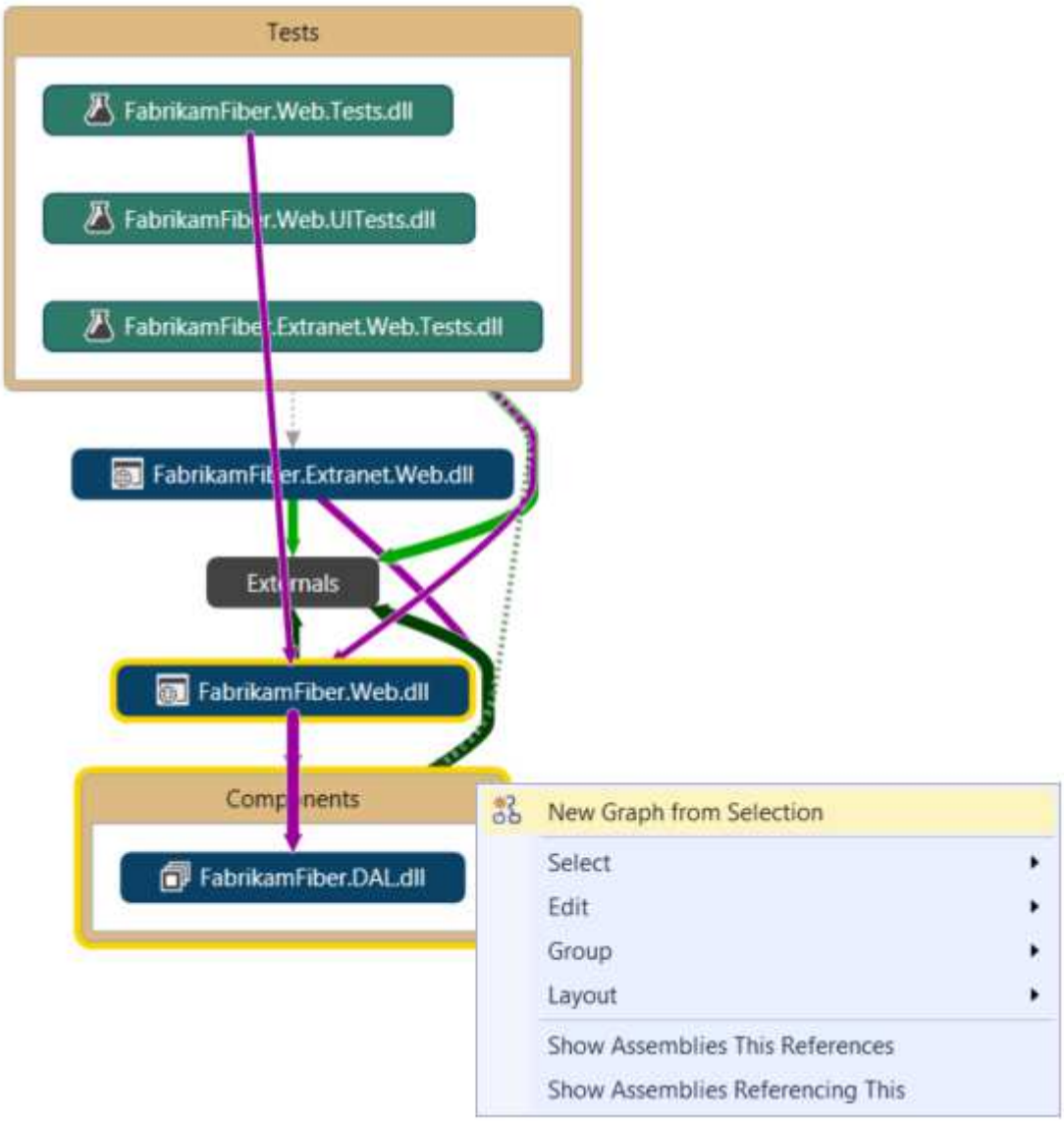
See CodeLens.

Design and modeling tools (Enterprise edition only)

Code maps and dependency graphs

In Visual Studio Enterprise, when you want to understand specific dependencies in your code, visualize them by creating code maps. You can then navigate these relationships by using the map, which appears next to your code. Code maps can also help you keep track of your place in the code while you work or debug code, so you'll read less code while you learn more about your code's design.

In this release, we made the shortcut menus for code elements and links much easier to use by grouping commands into sections related to selecting, editing, managing groups, and changing the layout of group contents. Notice also that test projects are displayed in a different style from other projects, and that we updated the icons for elements on the map to more appropriate versions.



Other improvements include:

- **Improved top-down diagrams**. For medium to large Visual Studio solutions, you can now use a simplified Architecture menu to get a more useful code maps for your solution. The assemblies of your solution are grouped by the solution folders, so you can see them in context and leverage the effort you've put in structuring the solution. You'll immediately see project and assembly references, and then the link types appear. In addition, the assemblies external to your solution are grouped in a more compact way.

- **Test Projects are styled differently and can be filtered**. You can now more easily and quickly identify test projects on map because they are styled differently. They can also be filtered out so that you can focus on the application's working code.

- **Simplified external dependency links**. Dependency links no longer represent the inheritance from System.Object, System.ValueType, System.Enum, and System.Delegate, which makes it easier to see external dependencies in your code map.

- **'Drill-in into dependency links' takes filters into account**. You get a useful, clear diagram when expanding it to understand the contributions to a dependency link. The diagram is less cluttered, and it takes into account the link filtering options you've selected.

- **Code elements are added to a code map with their context**. Because diagrams now appear with their context (up to the assembly and solution folder that you can filter out if required), you get more useful diagrams when dragging and dropping code elements from Solution Explorer, Class View, Object Browser; or when selecting elements in Solution Explorer and choosing Show on Code Map.
- **Get reactive code maps more quickly**. Drag and drop operations produce an immediate result, and the links between nodes are created much more quickly, without affecting subsequent user-initiated operations such as expanding a node or requesting more nodes. When you create code maps without building the solution, all the corner cases—such as when assemblies are not built—are now processed.
- **Skip rebuilding your solution.** Provides better performance when creating and editing diagrams.
- **Filter code element nodes and groups**. You can quickly unclutter your maps by showing or hiding code elements based on their category, as well as by grouping code elements by solution folders, assemblies, namespaces, project folders, and types.
- **Filter relationships to make diagrams easier to read**. Link filtering now also applies to cross group links, which makes working with the filter window less intrusive than it was in previous releases.
- **Create diagrams from the Class View and Object Browser**. Drag and drop files and assemblies into a new or an existing map from the Class View and Object Browser windows.

See Map dependencies across your solutions.

Other design and modeling changes in this release:

- **Layer diagrams**. Update these diagrams using Class View and Object Browser. To meet software design requirements, use layer diagrams to describe the desired dependencies for your software. Keep code consistent with this design by finding code that doesn't meet these constraints, and by validating future code with this baseline.
- **UML diagrams**. You can no longer create UML class diagrams and sequence diagrams from code. But you still create these diagrams using new UML elements.
- **Architecture Explorer**. You can no longer use Architecture Explorer to create diagrams. But you can still use Solution Explorer.

Visual Studio Extensibility Tools

It's never been easier to install the Visual Studio Extensibility Tools (VS SDK and templates) as they are now included as an optional component during setup. The Extensibility Tools allow developers to write extensions to customize and add features to Visual Studio. For more information about Visual Studio extensibility, see Visual Studio SDK

If you'd like to include the Extensibility Tools with your custom installation, you can find them under **Features / Common Tools / Visual Studio Extensibility Tools**. You can also install the Extensibility Tools at a later time by opening the **New Project** dialog and selecting the **Install Visual Studio Extensibility Tools** item under **Visual C# / Extensibility**.

Please give feedback

Why send feedback to the Visual Studio team? Because we take customer feedback seriously. In fact, we review each piece of feedback that comes into our feedback system. Your feedback drives a lot of what we do.

Send a smile

Telling us what you like helps us understand when we meet or exceed your expectations. When we are designing and implementing new features, we use data about the features you like to help with our design decisions. So, if you like a feature in Visual Studio, do tell us about it. It's easy and you can do it directly from within the IDE.

Just click the yellow smiley face on the title bar, tell us what you liked then click the **Send a smile** button.

That's it! We'll route your feedback to the correct team where they'll pat themselves on the back then quickly begin thinking of ways to delight you even more.

Send a frown

Hearing where we need to make improvements in the product helps us to manage our backlog by focusing first on the things that are most important to our customers. If there is something that's bugging you, tell us about it by using the **Send a Frown** feature from directly within the IDE. We've made this a super simple process too:

Click the yellow smiley face on the title bar, then click **Send a Frown**. Tell us what you did not like then click the Send a frown button. For more information, see Talk to Us.

Report crashes, hangs and performance issues

Sometimes, a quick note in a frown just isn't enough to convey the full impact of something that you do not like. For the times when you have a hang, crash or performance issue, you can easily share repro steps, crash dumps and trace files by using the dialog that's displayed after you send a frown.

First, send a frown as described above. On the dialog that pops up, you can tag your feedback with either one of the default tags or create your own tag. Tags help us route your feedback to the appropriate feature team. In the **Choose a category** drop down list, select the option that represents the issue you're reporting then follow the steps to reproduce the issue. Detailed steps on how to use Visual Studio to report feedback are also available. For more information, see Visual Studio Send a Smile Instructions.

Track your issue in Connect

If you'd like to track the status of your Visual Studio 2015 feedback, go to Connect and report the bug there. After reporting the bug, you'll be able to return to Connect to track its status.

See Also

Build cross-platform apps with Visual Studio Tools for Apache Cordova

Build cross-platform apps with Xamarin in Visual Studio
Build cross-platform apps with Visual C++
Generate unit tests for your code with IntelliTest
Work with multiple user accounts
Create custom window layouts
Perform quick actions with light bulbs
What's new for Application Lifecycle Management in Visual Studio 2015