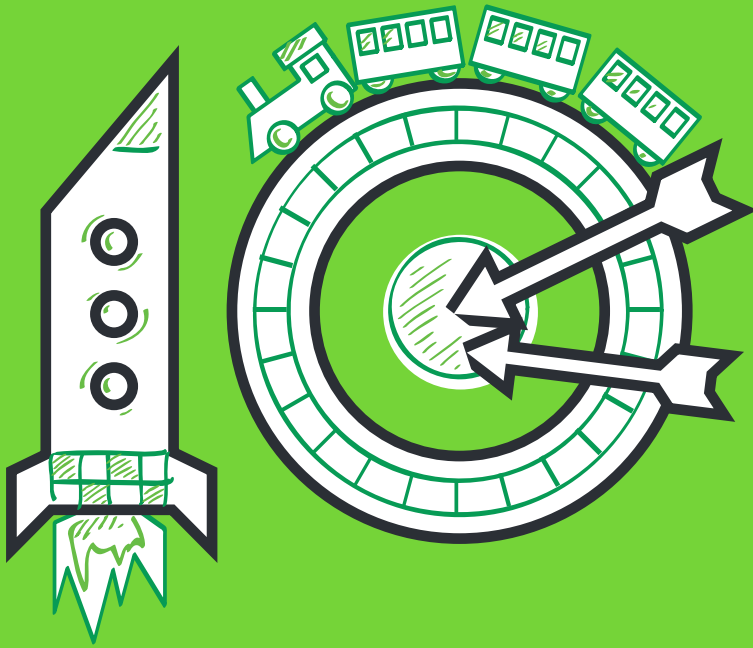


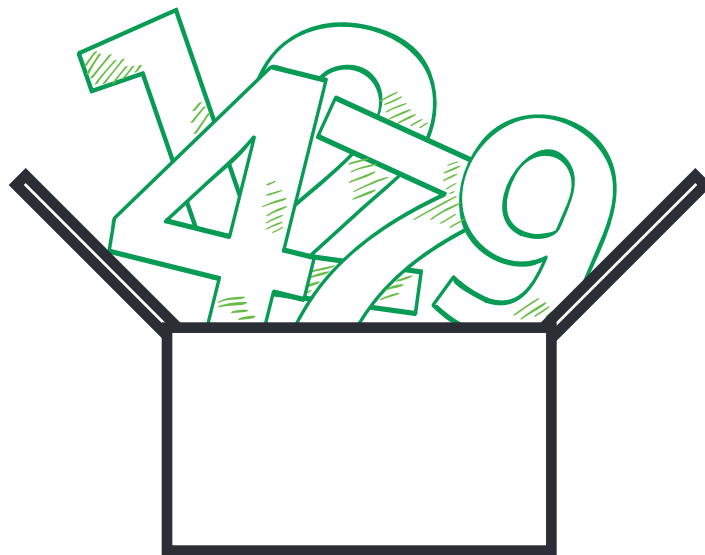
This may be out of date. [Read what's new here!](#)



# 10 GAME-CHANGING FEATURES IN VISUAL STUDIO 2013 FOR THE ASP.NET DEVELOPER

# CONTENTS

1. One ASP.NET	4
2. Security Configuration Made Simple	6
3. Scaffolding on Steroids	7
4. Say Goodbye to Design View and Hello to BrowserLink	8
5. Web Editor Intellisense—Not Just for HTML Tags Anymore	10
6. Bootstrap Templates	12
7. How Do I Get Started? Project_Readme to the Rescue	13
8. Integrated Azure Management	14
9. CodeLens Saves the Day	15
10. Integrated Git Source Control	16
11. How to Save Even More Time in Visual Studio 2013	17



# OVERVIEW



October 2013 brought a new release of Microsoft's Visual Studio. As with each Visual Studio update, there are framework enhancements and language enhancements for all project types and environments. In this edition, we see some of the most dramatic changes to ASP.NET since the introduction of the web development framework in .NET 1.0

This whitepaper will review 10 of the most important changes and explain what makes them so valuable for ASP.NET developers. You will learn why Visual Studio 2013 is the version of Visual Studio to upgrade to, no matter if you are currently running in version 2012, 2010, 2008 or 2005.

# 1. ONE ASP.NET

With the release of the ASP.NET MVC framework in 2009, web programming with .NET became something of a guessing game. At the beginning of any project, a choice would have to be made: Should one build with Web Forms or MVC? There have always been options to use the two project types together, but developers would combat the features of Visual Studio to be most efficient with their tools.

Starting with Visual Studio 2013, the project type guessing game is over. There is now only one web project type in Visual Studio.

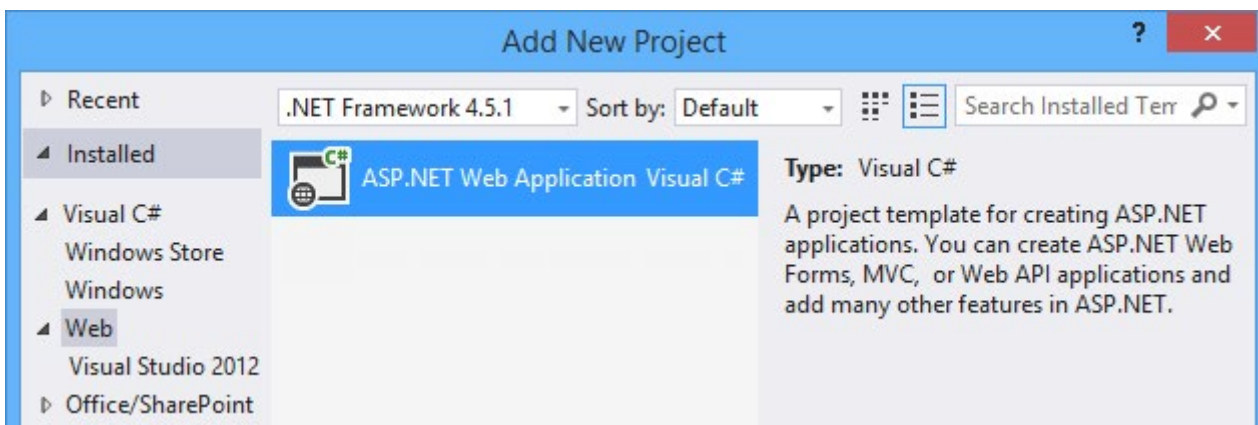


FIGURE 1 - ONE ASP.NET PROJECT TYPE

In this model, we have features and functionality that enable us to utilize the entire ASP.NET toolbox available to us. When you begin working with this project type, you will be prompted with additional configuration options to begin working with your project.

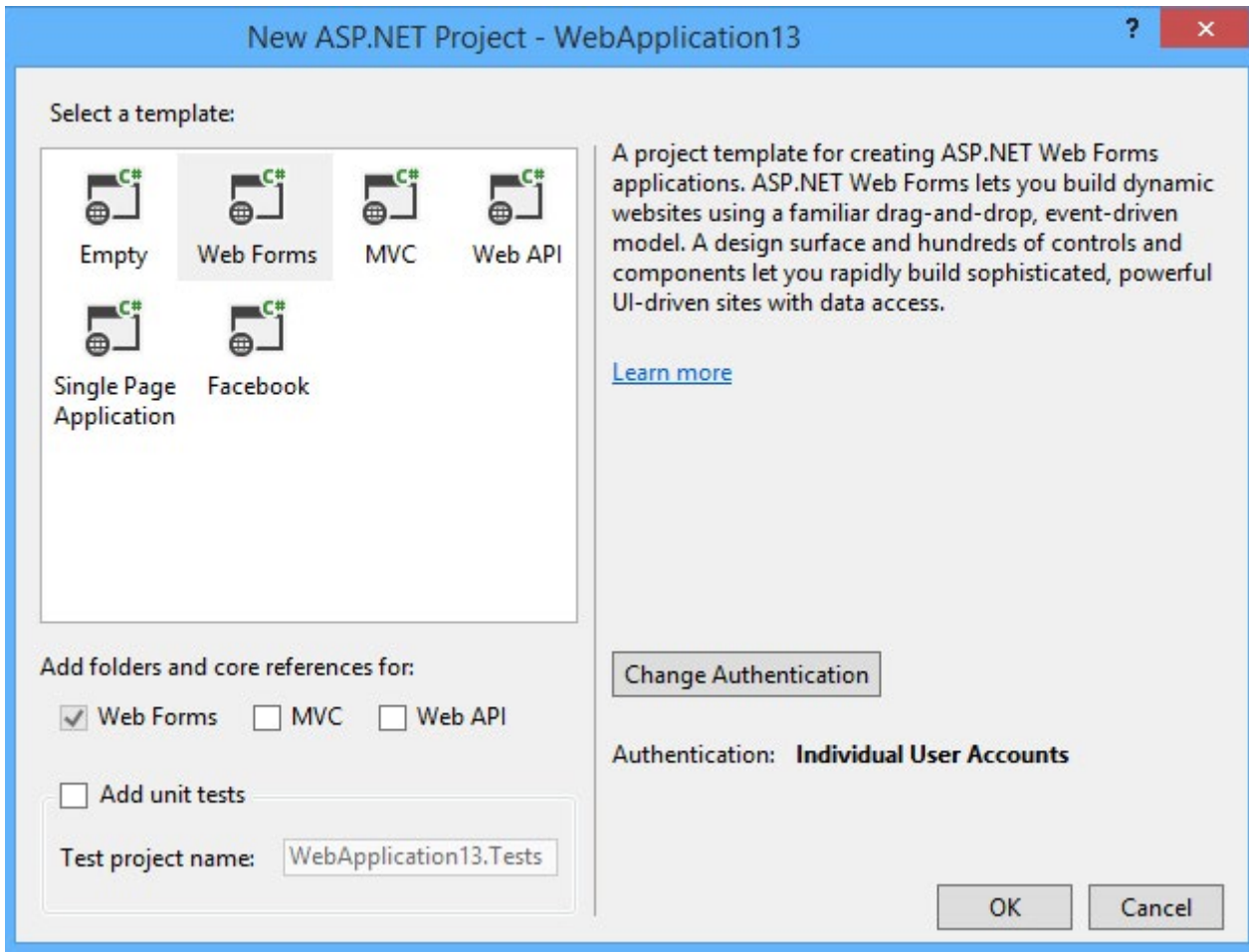


FIGURE 2 - PROJECT TEMPLATE CONFIGURATION

As you can see from the list of templates, we can choose to start with a standard Web Forms, MVC, or Web API project type. The other project types from the old MVC template dialog are still here. The interesting part is the checkboxes underneath the list of templates. Here, we can choose to add Web Forms references and folders to an MVC project or MVC references and folders to a Web Forms project. This is the project gateway to using more of these features in concert with each other.

## 2. SECURITY CONFIGURATION MADE SIMPLE

One of the trickier parts of ASP.NET is the configuration of authentication capabilities for your project. In previous versions, you needed a good understanding of the syntax of your web.config file to configure options properly. If you misconfigured something, there was no immediate feedback from Visual Studio. You only learned of a misconfiguration while debugging your application, when a white-screen error appeared indicating the error.

Starting with the One ASP.NET configuration dialog in this version of Visual Studio, we have a configuration wizard that will walk you through the security configuration and output the correct settings for you in your project files.

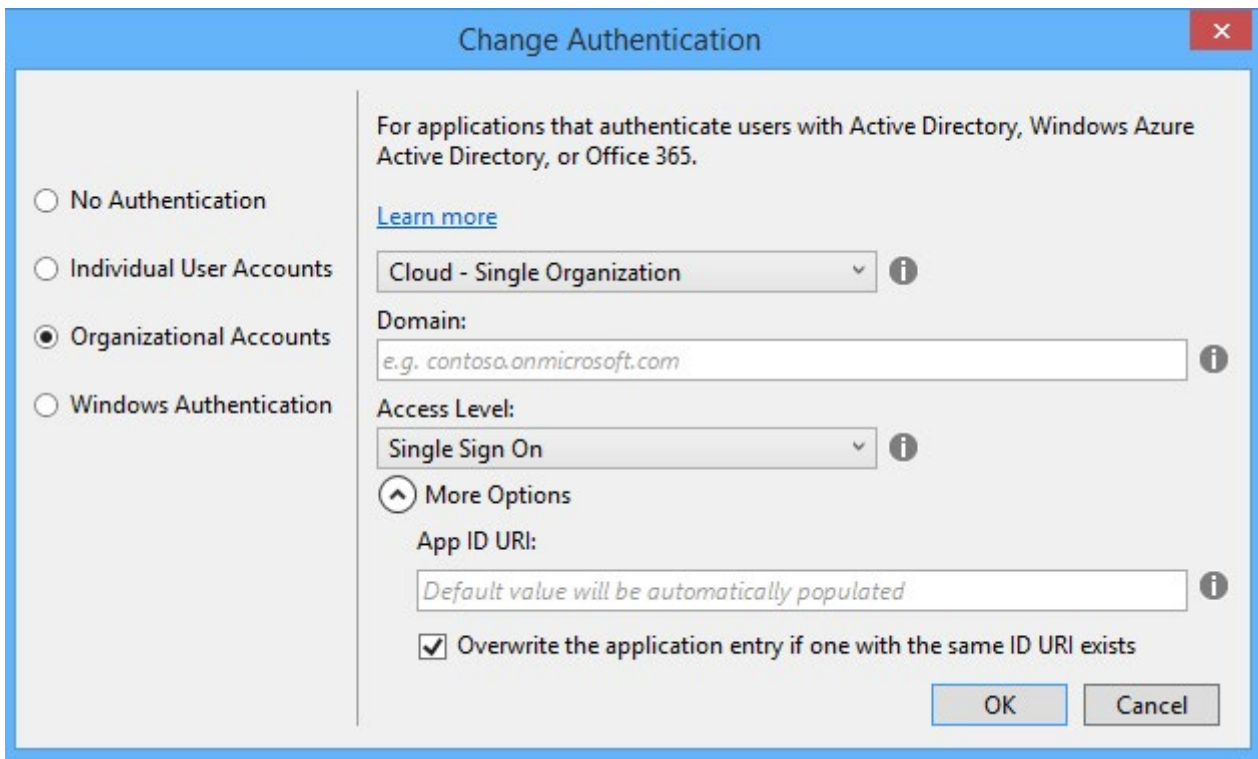


FIGURE 3 - AUTHENTICATION CONFIGURATION SCREEN

This wizard has steps configured to assist in configuring even the most complex of authentication scenarios—Single Sign On using Active Directory from a foreign domain. These first Visual Studio wizard steps will get your project started correctly and you will be able to easily build on those steps for the life of your application.

# 3. SCAFFOLDING ON STEROIDS

In previous versions of Visual Studio, the scaffolding capabilities of ASP.NET MVC were a huge benefit. You could point to an Entity Framework context, a collection in that context, and Visual Studio would generate from templates a set of views, a controller and appropriate actions to complete basic read and write operations. In this new version of the application, we have significant options available to us when we want

to scaffold out objects in our ASP.NET project. You can begin to scaffold objects with the new “Add–New Scaffolded Item” menu option.

This new option removes many of the ugly sub-menus and child menus that were confusing— not just to developers but Visual Studio integrators as well. How many options of “Add” items do we really need in this menu before it becomes unwieldy and annoying to consume?

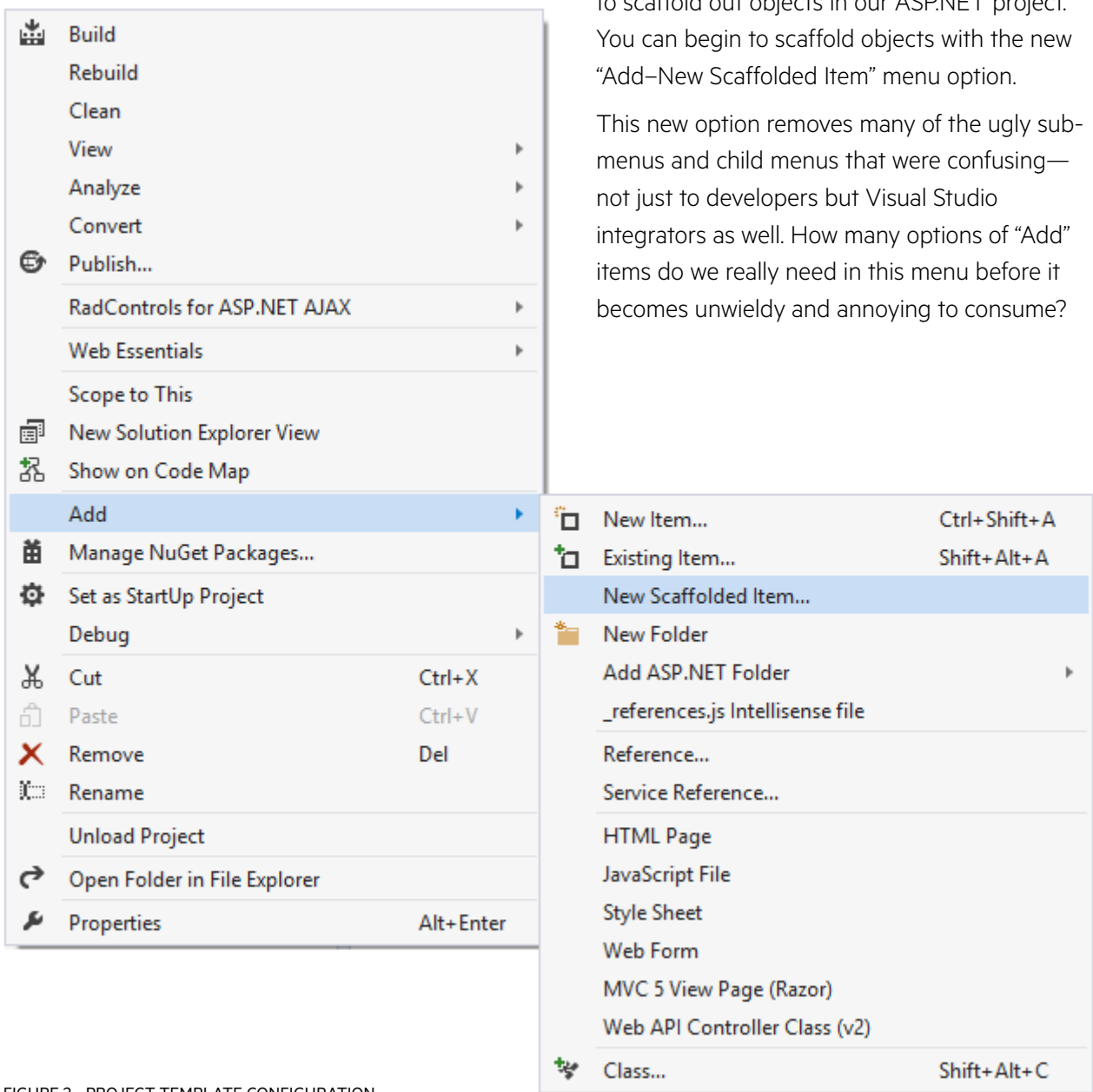


FIGURE 2 - PROJECT TEMPLATE CONFIGURATION

By choosing the “New Scaffolded Item” option, the following window opens, giving a large selection of great templates you can build on:

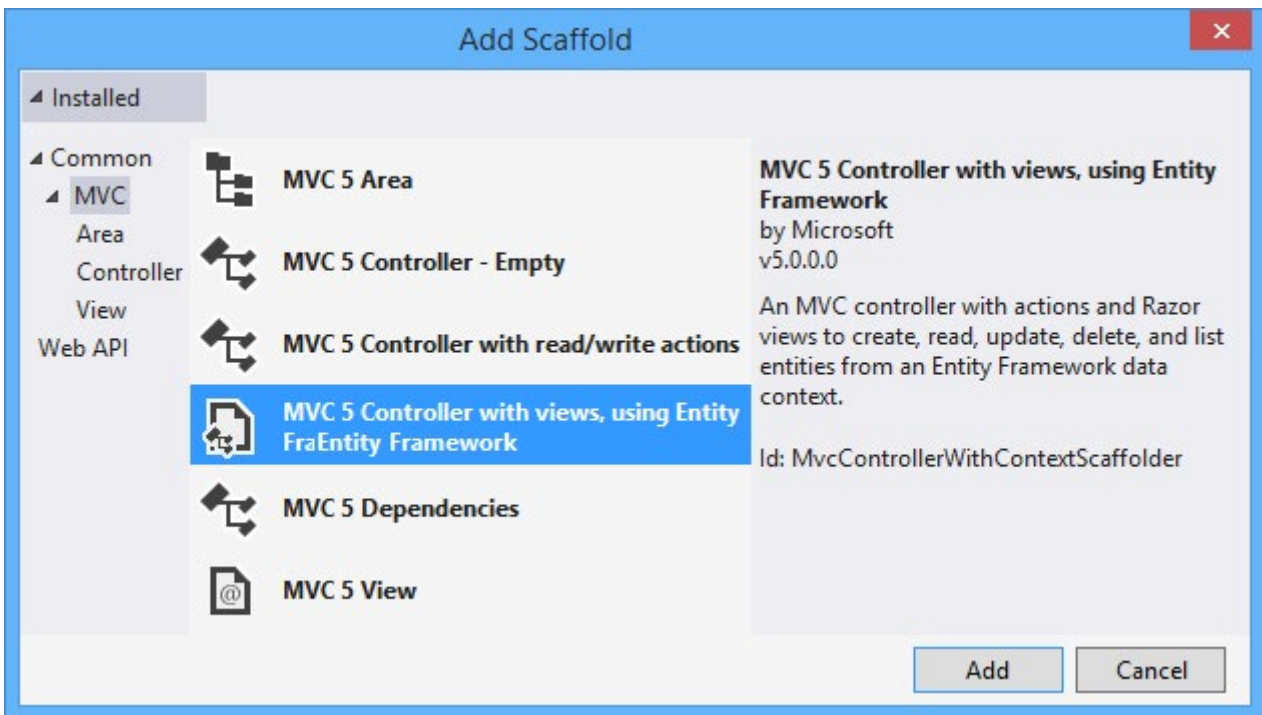


FIGURE 5 - SCAFFOLD OPTIONS

These options in the MVC selection should look familiar, except for one near the bottom: “MVC 5 Dependencies.” With this option, the MVC 5 framework can be added to a project that was started without the MVC folder structure and configuration. This option is significantly easier than the project file hacking and folder modifications required by prior versions of Visual Studio.

The other templates for MVC in this box lead to similar configuration windows that were present in prior versions of Visual Studio. You will be able to configure the naming for your objects and Entity Framework options appropriately.

New templates are being created and can be added to this dialogue. Microsoft wants you to be able to extend Visual Studio by using more templates. Check with your favorite software vendor to learn what templates are available to scaffold content.



# 4. SAY GOODBYE TO DESIGN VIEW AND HELLO TO BROWSERLINK

One of the coolest new features for web developers in Visual Studio is the Browser Link feature. This capability was code-named “Artery.” When you see it in action, you will understand the code-name reference, as this is now the way you will want to work with designing your web pages.

BrowserLink connects Visual Studio to any browser that is actively working with your web content, and provides a live connection from Visual Studio to that browser. Consider a situation where we need to support our website on many browsers and a mobile device or two:

We may end up looking at screens similar to what is presented in Figure 6. As we make changes to our page content in Visual Studio, it is difficult and tedious for developers to attempt to keep all of these browsers and emulators in sync. However, after changes to our content in Visual Studio we can click the BrowserLink refresh button or use the hotkey combination of Ctrl + Alt + Enter and force a refresh all of these browsers at once. This rapid return on browser feedback is a tremendous productivity boost for developers.

For an extra productivity boost, try installing the Web Essentials 2013 package. This collection of add-ons and extensions for Visual Studio adds extra features to BrowserLink, including Inspect and Design mode for MVC views. With this option, you can press Ctrl + Alt + D in your favorite browser and enter “Design Mode.” You can now start typing and changing content on the page, and BrowserLink will keep your MVC view’s content in sync. Text will immediately appear in the Visual Studio editor as you are typing it in the browser. How’s that for wysiwyg programming?

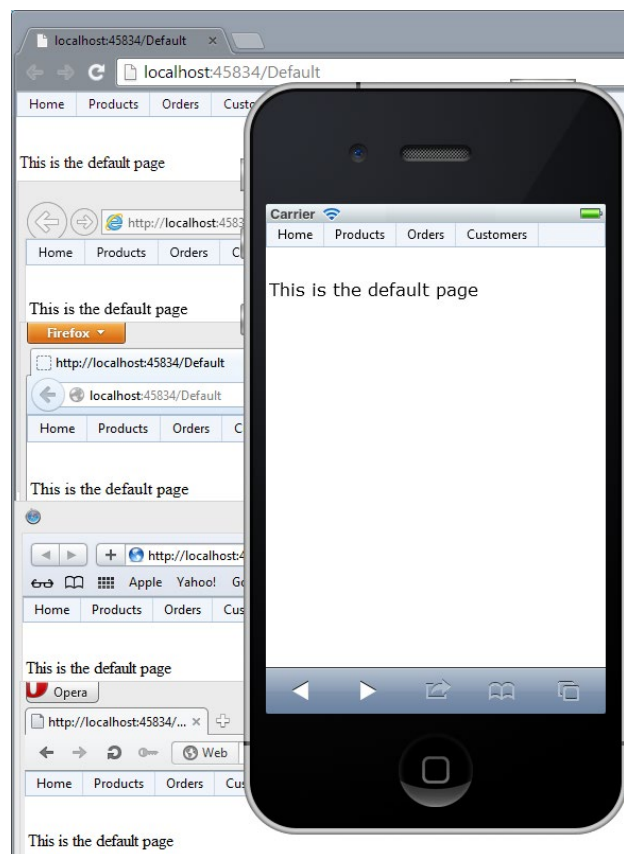


FIGURE 6 - TESTING ON MANY BROWSERS AT ONCE

BrowserLink has a number of extensions already available and a growing community of plugins and extensions. Check the BrowserLink section of [www.asp.net](http://www.asp.net) to see how you can further enhance your development experience.

# 5. WEB EDITOR INTELLISENSE IS NOT JUST FOR HTML TAGS ANYMORE

Prior versions of Visual Studio had a good text editor for HTML content. You could begin typing a tag name and it would give you IntelliSense options to complete the tag, some information about the attributes to place on that tag and of course add a default closing tag. These were simple options to add for almost any tag in HTML, but with the new version of Visual Studio, this functionality is improved ten-fold.

If you are working with an HTML element and start typing the CSS class attribute, Visual Studio will now parse all CSS sheets that are referenced and provide an IntelliSense drop-down of all of the classes that are currently defined in the CSS referenced by your page.

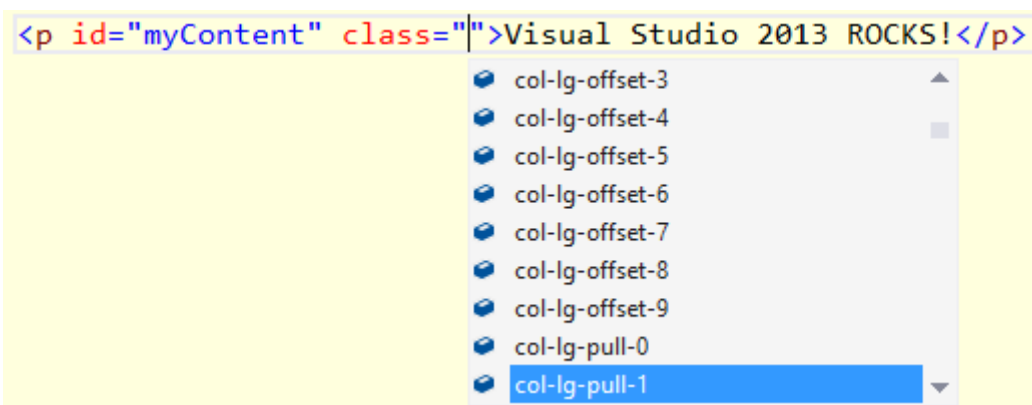


FIGURE 7 - CSS CLASS INTELLISENSE

In this example, we are referencing the Bootstrap framework, and all of the appropriate formatting classes from `Bootstrap.css` are available to us.

CSS is not the only IntelliSense in the browser for us; Visual Studio will also detect and present content for META tags. Microsoft has done us a favor here—and they do not limit the IntelliSense to features for Bing, Windows Phone or SharePoint. Check this out:

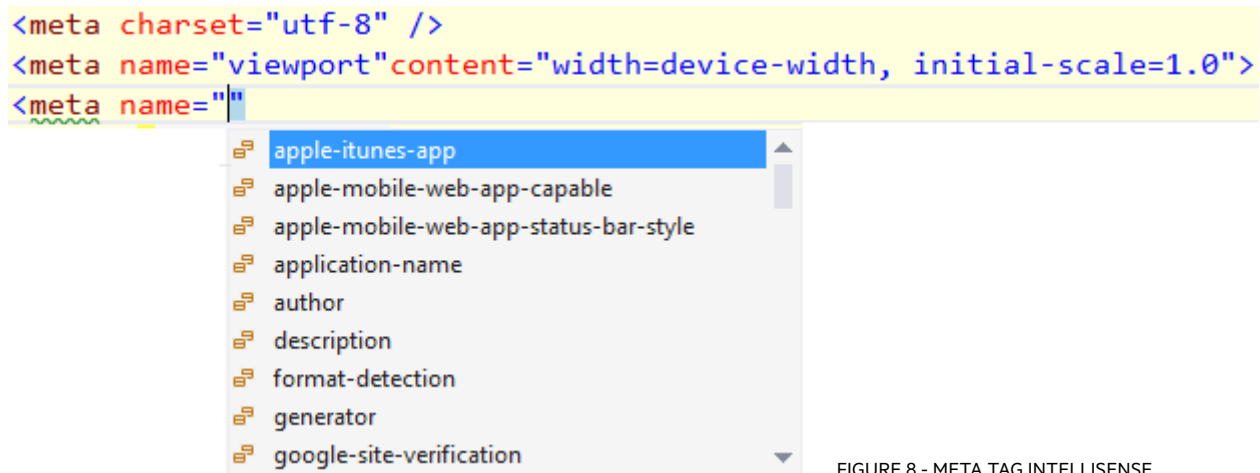


FIGURE 8 - META TAG INTELLISENSE

There are options to configure META elements for Apple, Google, Twitter and Microsoft. The content attribute of this tag has similar automatically detected options available after you choose a name attribute. This makes the configuration of options for these public sites that much easier to maintain and ensure you have configured properly.

What about JavaScript? In JavaScript, there are many features for IntelliSense available. Starting from the HTML markup, there are now autodetection and autocomplete options available for Angular and Knockout markup:

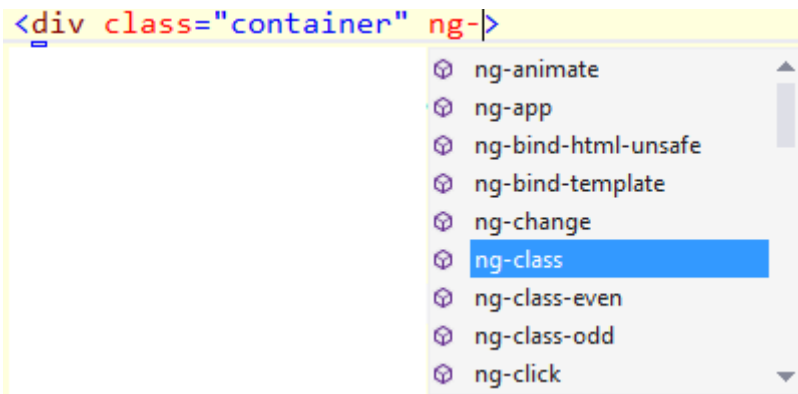


FIGURE 9 - AUTOCOMPLETE INTELLISENSE FOR ANGULARJS

The JavaScript editor has been completely re-written as well. You will find reference management and new “GoTo Definition” capabilities in the JavaScript editor. When editing JavaScript inline on an HTML page, you will find that the script knows about the entire DOM being presented. ASP.NET AJAX developers shouldn’t feel left out, as they get similar features when writing script inside of the ScriptManager object.

# 6. BOOTSTRAP TEMPLATES

The web of 2013 is evolving into a collection of sites that need to support not just multiple browsers but also multiple browser sizes and shapes on mobile and tablet devices. To meet this demand, Microsoft provided us with a default web site template that was responsive but difficult to build any serious content on.

In this version of Visual Studio, we now start ASP.NET projects with a Bootstrap-enabled template site. For those not familiar, Bootstrap (formerly Twitter Bootstrap) is a responsive CSS framework that contains markup and script that facilitate for simple and elegant application behavior on browsers of all sizes.

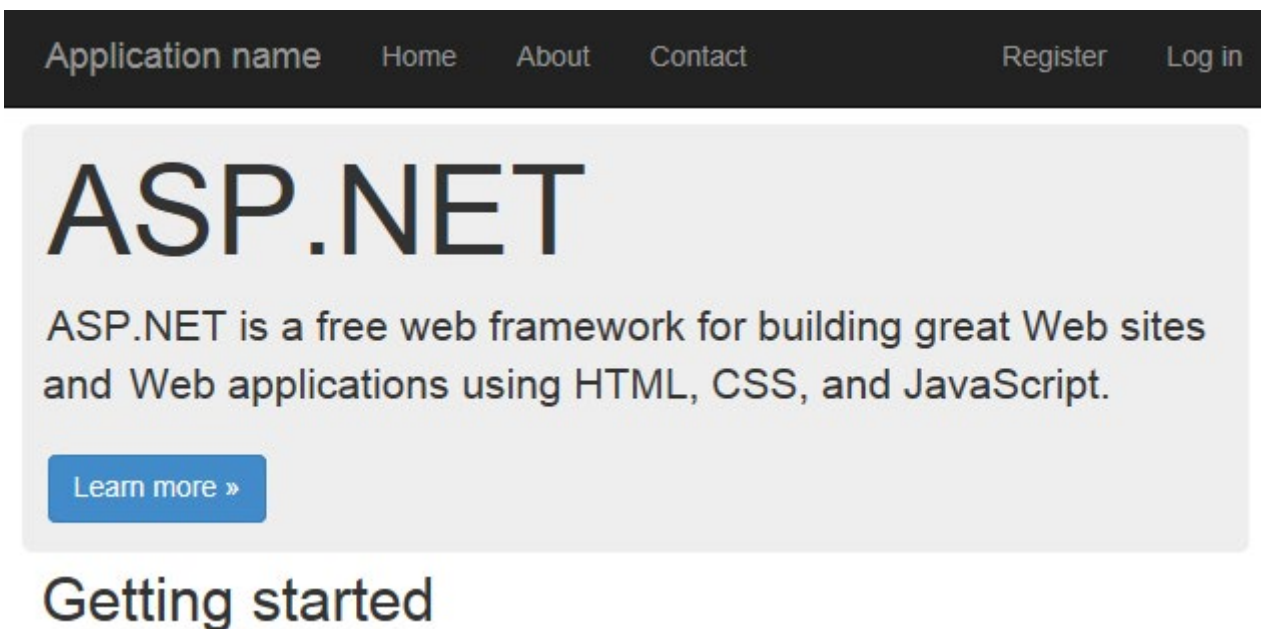


FIGURE 10 - BOOTSTRAP DEFAULT TEMPLATE

It does not matter which ASP.NET framework you choose to begin working with, all ASP.NET sites now begin with this framework. Using information from the [Bootstrap home page](#) you can quickly add elements that will be responsive and look great on browsers of all sizes. If you want to quickly apply a theme to your site, visit a CSS site like [Bootswatch](#) to download styles that will make your site look great.

# 7. HOW DO I GET STARTED? PROJECT\_README TO THE RESCUE!

It may seem simple to the outside observer, but because there is so much you can do with your ASP.NET project, it could be intimidating to the beginning developer. To help focus and give developers a list of places to go for more information and some simple directions, Microsoft added a Project\_Readme file to the root folder of new web projects.

<b>This application consists of:</b>	<b>CUSTOMIZE APP</b>	<b>DEPLOY</b>	<b>GET HELP</b>
Sample pages showing basic nav between Home, About, and Contact.	<a href="#">Get started with ASP.NET Web Forms</a>	<a href="#">Ensure your app is ready for production</a>	<a href="#">Get help</a>
Theming using <a href="#">Bootstrap</a>	<a href="#">Change the site's theme</a>	<a href="#">Windows Azure Hosting providers</a>	<a href="#">Get more templates</a>
	<a href="#">Add more libraries using NuGet</a>		
	<a href="#">Configure authentication</a>		
	<a href="#">Customize information about the website users</a>		

FIGURE 11 - PROJECT README - A GREAT START FOR NEW DEVELOPERS

This page will give you ideas and quick-start tasks like changing the site's theme, configuring authentication and even steps to help you deploy to Windows Azure. This is a huge step to make the very complex ASP.NET landscape much easier for developers to navigate and I think you'll find it very beneficial.

# 8. INTEGRATED AZURE MANAGEMENT

Microsoft makes a big deal about how easy it is to get started with Windows Azure. They've already simplified and made deployment to the cloud drop dead simple with the enhancements to that service, and they have brought those enhancements into Visual Studio.

With a new Windows Azure element in the Server Explorer, you can import your subscription information from Azure and manage your websites, SQL databases and Azure Mobile Services data:

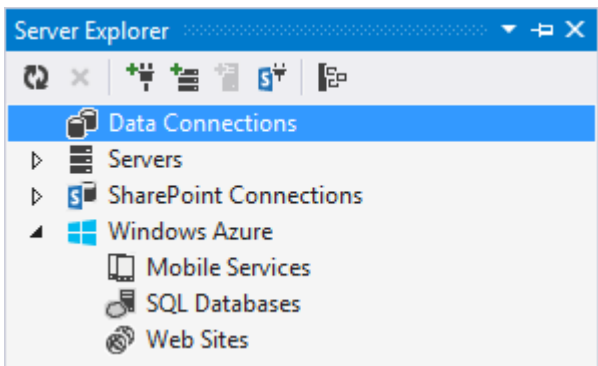


FIGURE 12 - WINDOWS AZURE MANAGEMENT ELEMENTS IN SERVER EXPLORER

Right click on the Windows Azure node and select “Import Subscription” to begin connecting Visual Studio to your Windows Azure account. There is a handy “Download Subscription File” link in the subsequent dialog window that you can use to acquire a settings file from the Windows Azure service. Once you have that file, you can browse for it and use it to configure Visual Studio.

With the connection to Azure established, you will be able to manage your tables of data in Mobile Services, work with JavaScript trigger scripts, open your SQL databases in SQL Object Explorer, manage the running state of your websites and watch the running logs of your websites. If you are running a production website in Windows Azure, you must activate this functionality to simplify your workflow.

# 9. CODELENS SAVES THE DAY!

A new feature that many beta testers did not fully understand is coming into its own. You may have heard about the “references” indicators that appeared in code during beta testing. These indicators are just the tip of the iceberg, with a TON of cool functionality hiding underneath them.

```
2 references | 1/1 passing  
public string Format(string toFormat)  
{  
    return toFormat.ToString();  
}
```

FIGURE 14 - CODELENS UNIT TEST INFORMATION

The references pop-up is similar and enables you to quickly see exactly how the method is used:

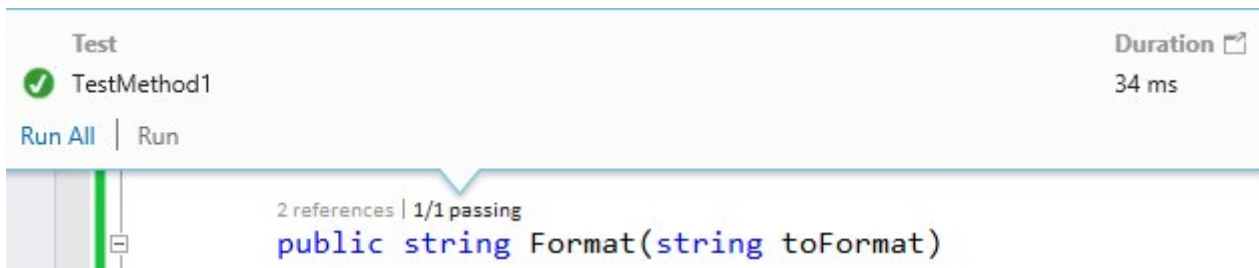


FIGURE 15 - CODELENS REFERENCES

It is very cool to be able to get such in-depth information from the code editor.

The real power of CodeLens is unleashed when you pair Visual Studio with Team Foundation Server. In this configuration, you will see information about the change history, who was the last editor of the method, the number of bugs and the number of work items that reference the method. These tools will aid your productivity and make the management of your projects significantly easier, all without leaving the comfort of your code editor.

If these indicators are distracting, you can easily turn off CodeLens from the Options—Text Editor window.

# 10. INTEGRATED GIT SOURCE CONTROL

Visual Studio has always favored Microsoft's Source Safe and Team Foundation Server for source control. Third party providers would have to adhere to an ugly interface to integrate and function inside of Visual Studio. If your provider did not have similar functionality to Source Safe, your source control provider would feel clumsy and awkward in Visual Studio. In particular, Git always felt a little goofy to use in Visual Studio.

In the service packs released for Visual Studio 2012, Microsoft began to embrace Git and added tooling support. Now, in Visual Studio 2013, Git is available for all users to work with.

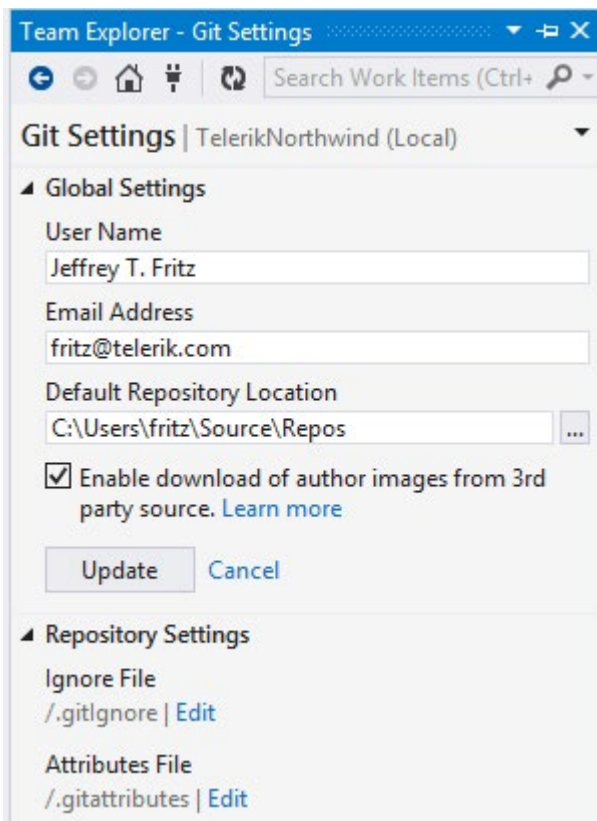


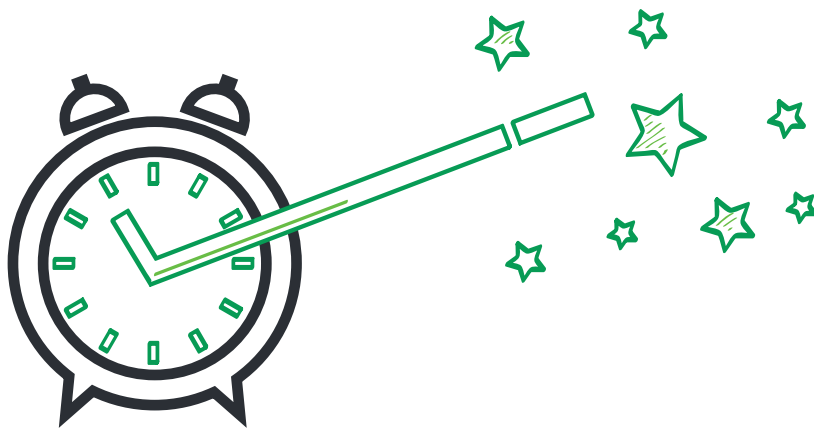
FIGURE 16 - GIT SETTINGS IN TEAM EXPLORER

Not only does the Team Explorer integrate cleanly with Git, the configuration options enable you to extract information about the authors in your project from third party sources. You will get photos and names of your co-authors when you review changesets.

Visual Studio is also adapted to the Git local commit and push changes to other locations. The configuration and changes within Visual Studio, to adhere to the workflow that developers are accustomed to when using a distributed source control system, are a welcome addition. Independent web developers and open-source developers will find this as a very welcome addition to the standard Visual Studio toolbox.



# CONCLUSION



Visual Studio 2013 is a massive leap forward for ASP.NET developers. We have returned to the original days of ASP.NET with a single web application project model and we now have significant new tools and features to help you keep pace with the trends and technologies of the evolving web.

[Download Visual Studio 2013](#)

# Try **UI for ASP.NET AJAX** and **UI for ASP.NET MVC** for 30 days

With over 150 UI controls and thousands of scenarios covered out-of-the-box, you can leave the complexities of UI to us and focus on the business logic of your apps

Try ASP.NET AJAX controls

Try ASP.NET MVC extensions



## Telerik

Develop experiences