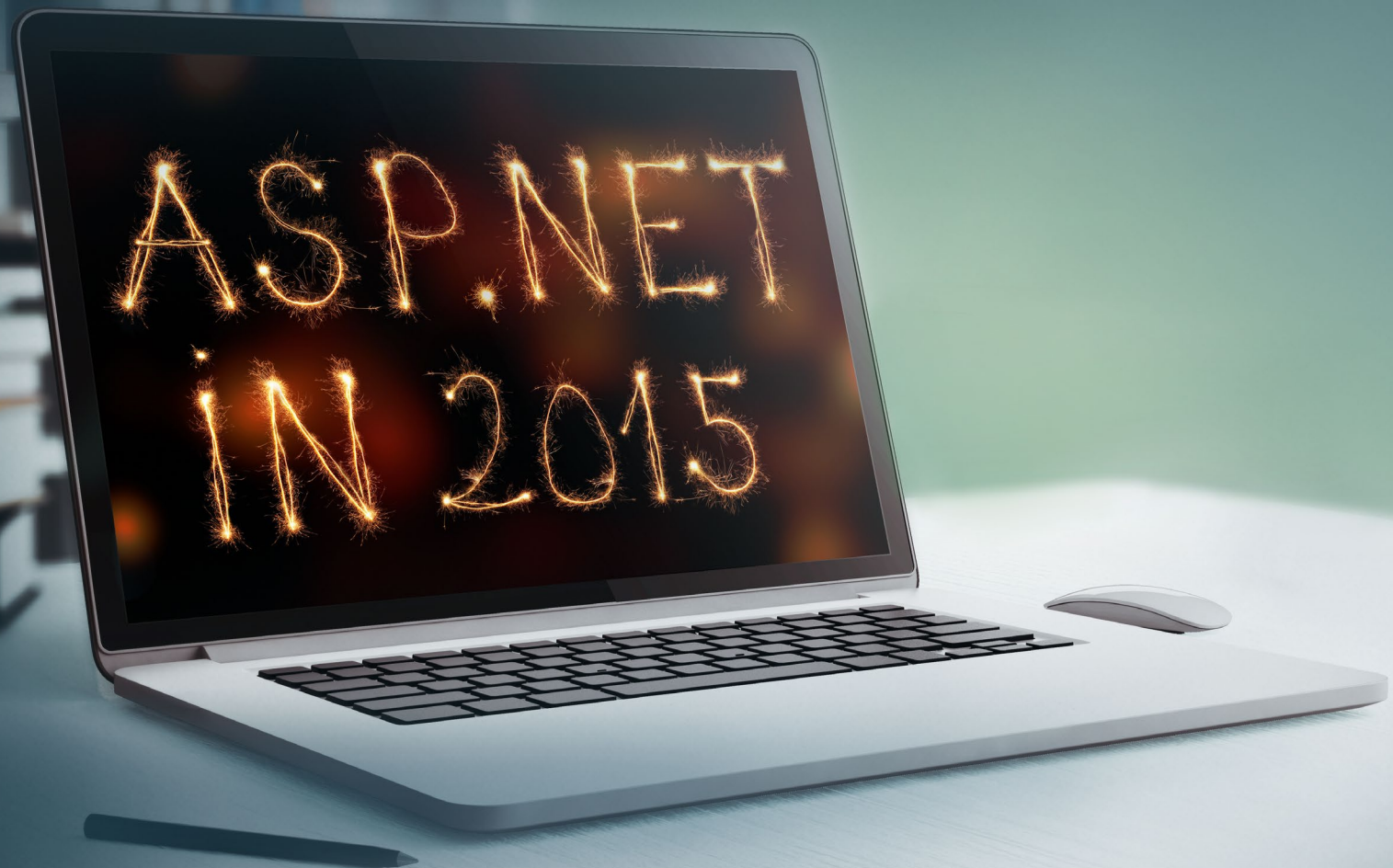


20 BURNING QUESTIONS ANSWERED



Are you an ASP.NET web developer? Hearing a lot of buzz lately, but honestly a little frazzled by all that's going on? Major changes often bring initial uncertainty and your hesitancy is only normal.

However, allow me to prove to you why I believe this is one of the best times to be an ASP.NET developer. No fluff—just a developer-to-developer breakdown of what's in store in the form of Frequently Asked Questions. Let's take a step back and understand the whole picture. The future of .NET is awesome and you'll be glad that you're a part of this.

Question 1:

Whoa—I heard .NET was open sourced?



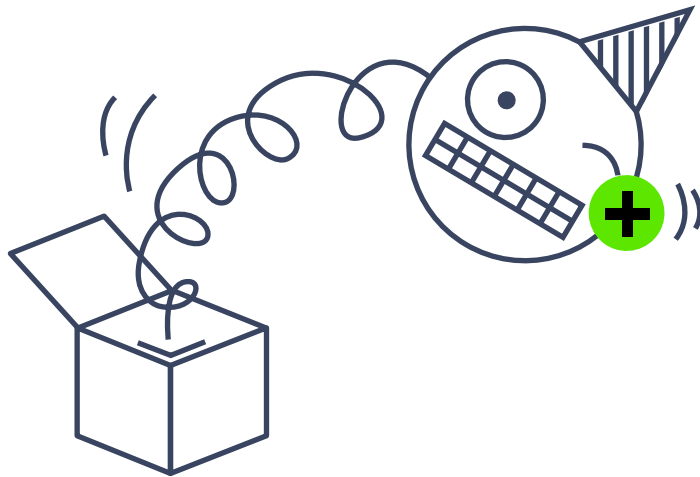
Answer:

November 12, 2014 will be marked as a day of monumental shift in Microsoft development stacks. At the [ConnectO](#) event in NYC, it was [announced](#) that the core of your beloved .NET Framework is now entirely open sourced and usable under an [MIT license](#). This will include everything needed to execute .NET code, including the

Common Language Runtime ([CLR](#)), Just-In-Time Compiler ([JIT](#)), Garbage Collector ([GC](#)) and core .NET base class libraries. So [grab the source code](#), use it, build it or fork it, with a lot of freedom. Microsoft is committed to accepting meaningful community contributions, and you can be assured of the quality of the .NET base classes.

Question 2:

Doesn't open source mean lower quality and greater risk?



Answer:

First of all, let's understand and open our minds to Open Source Software (OSS). Primarily, it means that software source code is available publicly and usable (to study/change/distribute) under a variety of license constraints. One fundamental trait of OSS is that it is often developed in a collaborative manner, thus leveraging continuous feedback. Don't like something? Pull down the source code and make changes to fit your needs.

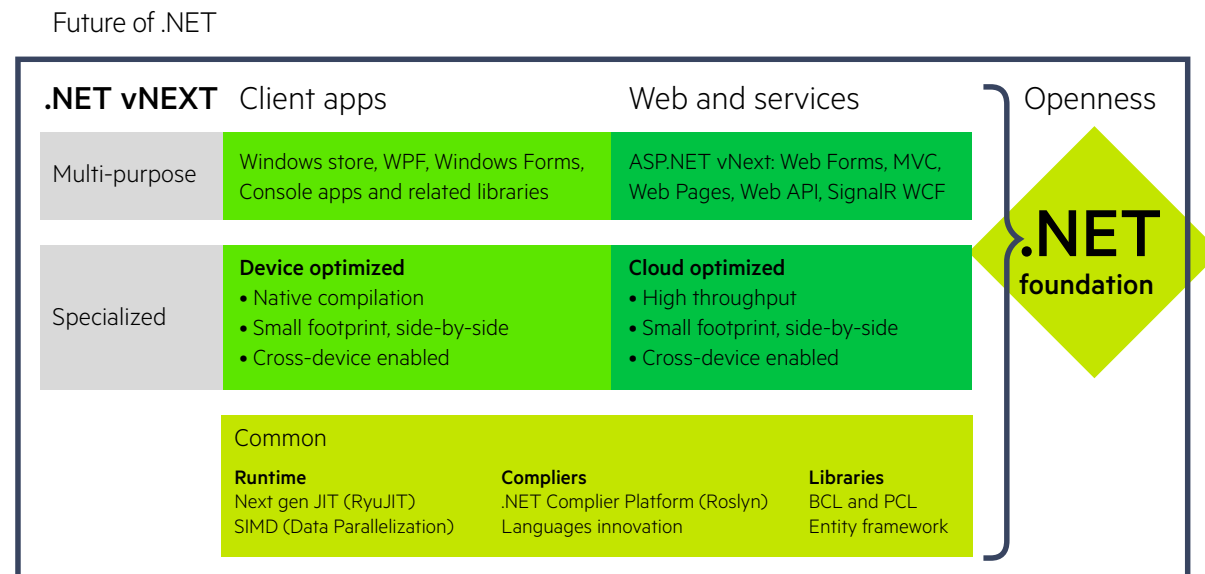
OSS also benefits tremendously from developer community involvement—meaningful changes are selectively accepted back into the primary source code and made available to all users. Now, if you believe that OSS is tricky or it's just a marketing gimmick, it's time for a serious mindset change and a look at the bigger goal. Yes, enterprises do need to consider the legal implications of OSS in their development stacks, but the overall benefits of open source far outweigh anything else. As for open sourced .NET, Microsoft will be the ultimate gatekeeper, so there will be no changes to quality or risk. Embrace OSS whole-heartedly—it's a great thing!

Question 3:

Does this mean I have to change the way I use .NET Framework?

Answer:

Yes and no. Let's take a 10,000-foot view:



The representation of .NET vNext above depicts how the framework is moving forward—it is very familiar and different at the same time. Fundamentally, it is the role and usage of .NET that is changing to offer increased flexibility. .NET used to be this behemoth—serving desktop, web or mobile app development and servers installations with equal footing. Moving forward, .NET will be much more specialized serving cloud, device and servers with optimizations.

Powering .NET will be a common set of features (runtime, compilers and base libraries), but you get to pick and choose exactly what you want to use. Are you used to waiting on IT or have an elaborate process before you upgrade the .NET Framework on your server? Well, now you can roll in the .NET framework self-contained in your apps and have multiple versions side-by-side. Fun!

Question 4:

Any goodies coming with this new .NET?

Answer:

The .NET Framework vNext packs a punch when it comes to some new language and compiler level enhancements. For starters, there is [Roslyn](#)—the .NET compiler platform. Roslyn allows for innovative C# or VB compilations in the cloud and has loads of benefits inside IDE (Integrated Development Environments) like Visual Studio, as well as third-party integrations. Windows Store apps benefit from [.NET Native](#) ahead-of-time compilation, resulting in near-metal performance with quicker app start up times and lower memory footprint. Desktop and server apps benefit from next-generation 64-bit [RyuJIT](#) (Just-In-Time) compilers. All this goodness for .NET applies to all application types and a leading to a convergence of development experiences.

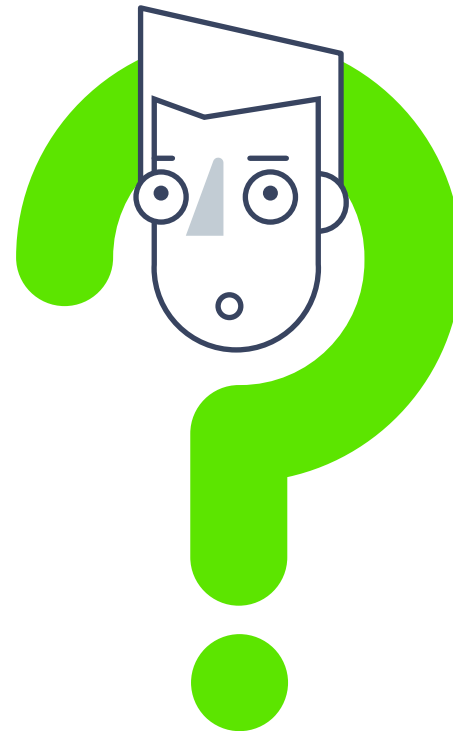


Question 5:

I keep hearing about the .NET Foundation. What is it exactly?

Answer:

Back in BUILD 2014 conference, [Scott Guthrie](#) announced the .NET Foundation. Quoting the [Home page](#): “The .NET Foundation was created as an independent forum to foster open development and collaboration around the growing collection of open source technologies for .NET”. The Foundation has since garnered a lot of support to host an impressive array of [OSS projects](#), both from Microsoft and supporting partners, and serves as the defacto stewardship home for open source repositories for .NET. This includes the next generation of ASP.NET.

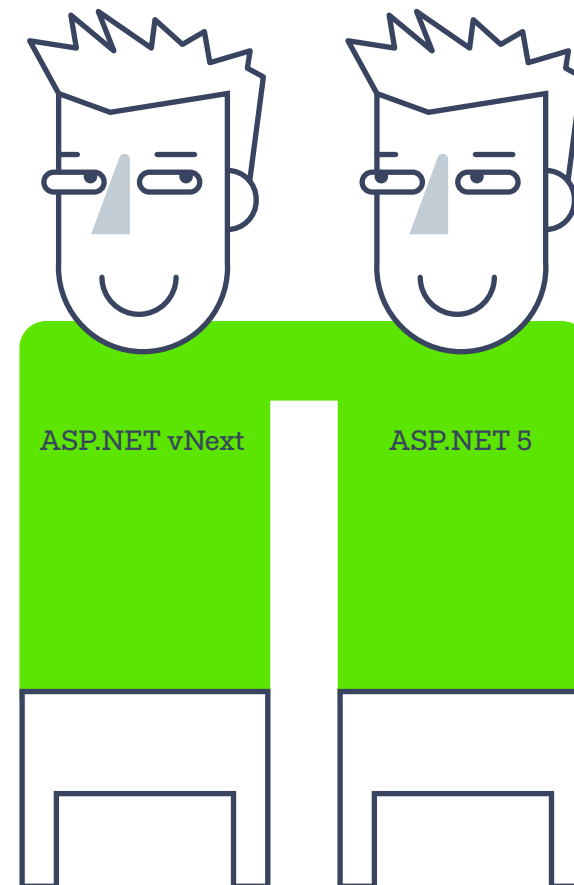


Question 6:

I've heard ASP.NET vNext and ASP.NET 5. What's the difference?

Answer:

Thankfully, they are the same. Plans for the next version of ASP.NET platform were announced a while back and since then, it has commonly been referred to as ASP.NET vNext. However, with the next release being somewhat solidified, the official name of ASP.NET vNext is ASP.NET 5. That's what you should expect to see in official communication moving forward.



Question 7:

So is ASP.NET vNext going to be open sourced as well?

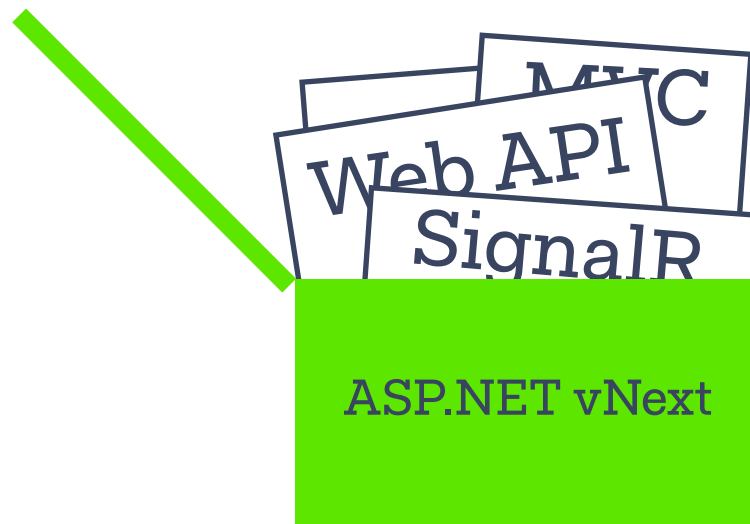


Answer:

Yes. Per the trend, ASP.NET vNext is entirely open sourced. You need not have any special insider access—go look at the source code, build, change, pull or fork. The GitHub home for ASP.NET vNext is <https://github.com/aspnet/home>. Don't fret though; no one is moving your cheese, just offering more flexibility.

Question 8:

What are the frameworks included in ASP.NET vNext?



Answer:

ASP.NET vNext sees the convergence of several frameworks into a unified programming model for MVC 6. This includes the latest MVC, WebPages, Web API, SignalR and Entity Framework. You can now have a single controller that returns both MVC views and formatted Web API responses, on the same HTTP verb.

Question 9:

I've heard ASP.NET has been broken up into pieces?



Answer:

Yes, the new ASP.NET framework is completely modular—you pick the pieces you need and get them through [NuGet](#). Don't worry though: you can still use the whole framework just as you do today. The future of [.NET on the server](#) looks interesting, to say the least. The ASP.NET framework has been written from ground up and is now a lean .NET stack for building modern web applications. You get to optimize your web applications for running in the cloud as well as on-premise, and you get to pick the framework pieces you need with modular components.

Question 10:

Is anything changing in the ASP.NET Runtime?

Answer:

Let's take a look at how the new age ASP.NET web applications actually run. They depend, of course, on the underlying .NET Framework, but with a few interesting changes. In the past, the .NET Framework was delivered as a single, all-encompassing installation, which continually grew in size. Sure your Windows server with .NET Framework could run any type of application, but ASP.NET web applications didn't actually require all of the framework pieces. This also made updating .NET Frameworks on server a non-trivial task, requiring lots of planning and processes.

ASP.NET 5 is meant to offer flexibility by being able to run on three runtimes:

1. Full .NET CLR

The full .NET framework as a runtime is the default for ASP.NET 5 projects in Visual Studio. It provides the entire framework's features and is your best option for backwards compatibility. This is the same as what you've been doing before ASP.NET 5 and you can keep doing so moving forward.

2. Core CLR (cloud-optimized runtime)

The Core .NET CLR is a lean and modular runtime for ASP.NET 5 projects. This .NET core has been refactored into components, so you get to choose only what is needed through NuGet packages. The Core .NET CLR is about 11 MB, compared to ~200 MB for the full .NET CLR. The Core CLR can be deployed with your ASP.NET web application and different versions of the CLR can run side-by-side. See the benefits for your Windows server?

3. Cross-Platform CLR

Microsoft will release a mono-based cross-platform runtime for Linux and Mac OS X, which will enable you to develop and run ASP.NET applications just like on Windows. This is the same Core CLR, just running outside of Windows.

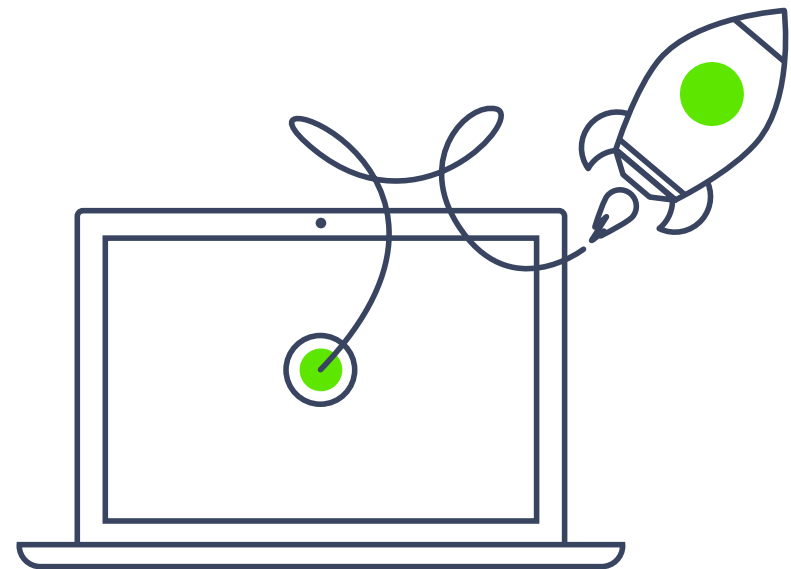
You can change the runtime for your ASP.NET 5 web applications through easy project configuration.

Question 11:

Wait. Are you telling me that .NET applications can run outside of Windows?

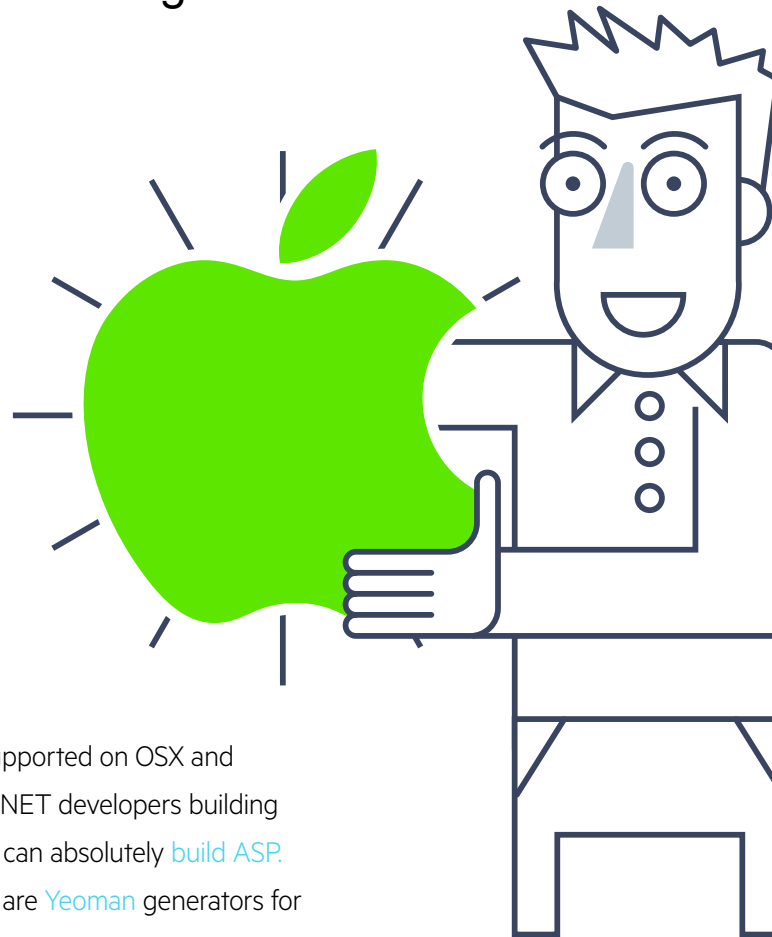
Answer:

We do not live in a silo and Windows isn't the only OS in the world—this mindset change within Microsoft is very evident in what's next for the .NET Framework. Official distributions of .NET Core will be available for Linux and OS X. Let this sink in, because it is a big deal. Cross-platform developers are now most welcome to use the .NET Framework. Web, desktop, cloud or mobile development on almost any platform can now be targeted with the .NET framework. You can build native ASP.NET web applications on a Mac. You can build native iOS or Android applications using C#, along with Windows counterparts. And you can build Hybrid single-codebase cross-platform mobile apps using plain HTML5/CSS/JS and the [Apache Cordova](#) open source framework. This is a huge and welcome change and that has the potential to spread the .NET love among non-Microsoft developers.



Question 12:

So I can build an ASP.NET application today natively on a Mac? This is mind blowing!



Answer:

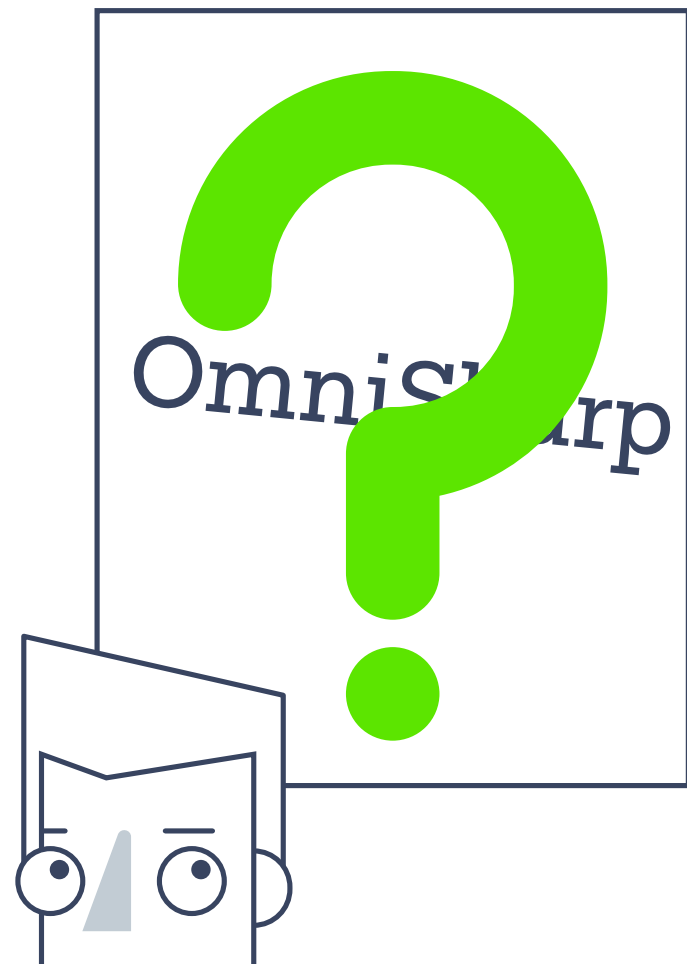
Yep, believe it. With the .NET core being supported on OSX and Linux, it does open up possibilities for ASP.NET developers building applications that target the .NET core. You can absolutely [build ASP.NET applications natively on a Mac](#). There are [Yeoman](#) generators for ASP.NET and you can use the [CLI](#) (Command Line Tools) to run ASP.NET natively on OSX.

Question 13:

I keep hearing about OmniSharp—what is it?

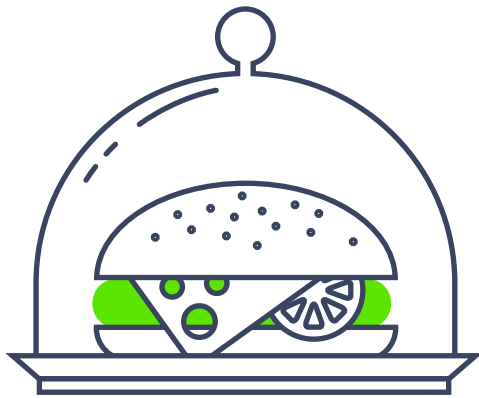
Answer:

[OmniSharp](#) is a very cool initiative to bring .NET development goodness to any platform or code editor. It is open sourced, as expected, but also has several ASP.NET team members contributing. OmniSharp is a family of projects, each with one goal: to enable great .NET development in YOUR editor of choice. This includes Sublime Text, Atom, Emacs, Brackets, VIM and others. You get true VS-like intellisense, thanks to the power of Roslyn and other open source interpreters. [C# in Sublime Text](#) is a first-class citizen, providing all the features you expect from a rich editor. You can develop your next ASP.NET web application almost anywhere—the .NET developer world just got very interesting.



Question 14:

Amidst all these cool sounding changes, are ASP.NET WebForms dying?
You are moving my cheese, aren't you?



Answer:

With all this talk of modern MVC and ASP.NET frameworks, it is easy to get worried if you are developing web applications using ASP.NET Web Forms. Relax and breathe--it's not going anywhere. Web Forms continues to be one of the easiest, most economical ways of developing web applications, and it will continue to be an essential part of the .NET web development platform.

In fact, Microsoft remains focused on adding new features to Web Forms to improve the development experience and keep the technology up-to-date with web practices. Your apps that currently use Web Forms, MVC 5, Web API 2, SignalR 2, Web Pages 3 or Entity Framework 6 will be fully supported on the new framework without modification. However, you will most likely need to use the full .NET CLR to run legacy apps because only this CLR provides full compatibility with earlier versions.

Question 15:

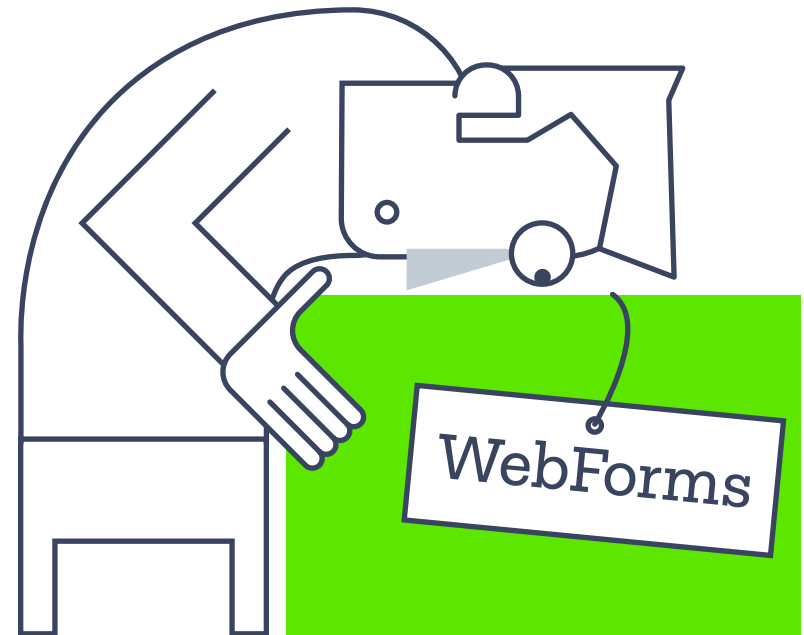
Anything new in WebForms? Would I have to update my existing applications?

Answer:

Web Forms 4.6, included in ASP.NET 5, has the following new features:

- HTTP 2: New HTTP processing pipeline
- Async model binding: For strongly typed data handling
- Roslyn CodeDOM compilers: For faster compilation of code-behind files

Your existing Web Forms ASP.NET web applications will continue to run without any modification on IIS with .NET 4.6. Let me restate: no changes are required for your existing Web Forms applications to run with latest .NET framework. You cannot, however, use Web Forms apps with the cloud-optimized runtime—rather you'll need to use the full .NET CLR as you always have. Learn more about Web Forms 4.6 from this [Channel 9 video](#).



Question 16:

Anything else new in ASP.NET 5?



Answer:

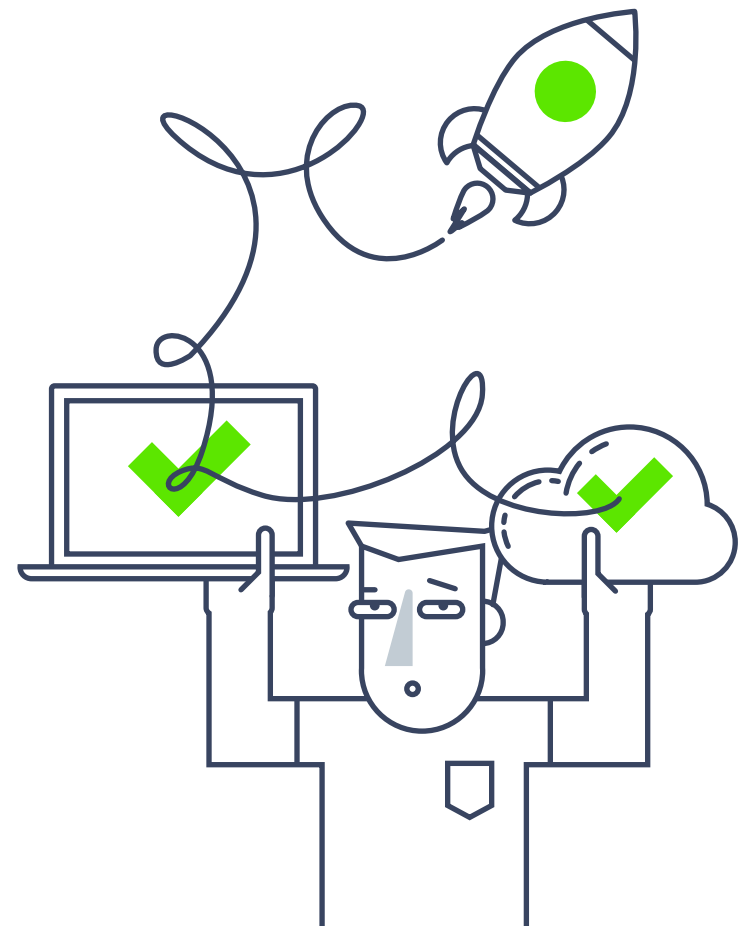
There is a brand new HTTP processing pipeline with terrific throughput. The new [KRuntime](#) is the core of ASP.NET vNext. It comes with a built-in [Version](#) and [Package](#) manager along with loads of other tooling. ASP.NET MVC syntax gets these little magical things called [Tag Helpers](#). Learn about them, since they are quite awesome. ASP.NET web applications can be hosted in traditional IIS or outside in their own processes.

Question 17:

So, future ASP.NET applications can run on-premise, as well as in the cloud?

Answer:

Yes. You can self-host ASP.NET applications in IIS inside your enterprise infrastructure, like you always have. You can target the whole full .NET CLR and keep doing things the same way. Alternatively, you can target the Core CLR, which is cloud optimized—you get to pick the framework pieces to run on your server. This allows for easier .NET Framework upgrades, because different apps can run in isolation. ASP.NET vNext is also cloud-ready. Features such as session state and caching provide consistent APIs and adjust themselves with behavior well-suited for cloud hosting versus traditional servers. Dependency injection is also supported throughout ASP.NET—plug and play to your heart's desire.



Question 18:

Are there any tooling improvements for ASP.NET Developers?

Answer:

There are lots of tooling improvements for .NET developers inside Visual Studio, but the big news about a brand new Visual Studio SKU trumps everything else. Remember the VS Express editions? They were free for sure, but you could only develop one type of application with each specific VS Express editions. Now, there is the new [Visual Studio Community Edition](#) – the one SKU to rule them all. It's free, and you can develop any type of .NET application with it—web, cloud, desktop or mobile. It is a full-featured IDE that supports advanced features like plugins and extensions. And, you can develop modern ASP.NET 5 web applications with VS Community Edition.



The best news around tooling comes in the form of improved debugging for ASP.NET web application developers. Let's say you are building a complex enterprise web application. Every time you change any .NET (C#/VB) code in the code behind files, you go through a full build step for your browser to pick up the changes today, right? Well, with ASP.NET vNext, simply make your code changes, save your files and refresh your browser to see the changes. And I'm not talking HTML or other markup, but C#/VB server-side code. Thanks to JIT compilation from Roslyn, the wonderful .NET compiler platform, compilation is on-the-fly and a whole lot faster.

In 2015, you also get to use [Visual Studio 2015](#) with all its [bells and whistles](#), including integrated package managers/builds tools for client-side development in [Bower/NPM/Grunt](#) support. You also get to use the [new features in C# 6.0](#) for your server-side comfort

Telerik Control Suites

.NET developers have always loved Telerik® Controls. To provide an idea of scale, the [Telerik DevCraft™](#) bundle (which includes all our .NET tooling) is used by 1.4 Million developers. That's a lot of love and we are honored. As .NET evolves into its next generation, so will we. You will find us adopting the best features, performance enhancements and innovating more with specialized controls for each platform, as well as building on the already present cross-platform story. Here are two of our major toolsets around ASP.NET that help you build awesome web applications for the next generation:

- [Telerik UI for ASP.NET AJAX](#): Rich WebForms framework with 80+ controls and support for every device
- [Telerik UI for ASP.NET MVC](#): 70+ ASP.NET MVC mobile-ready components for building responsive apps powered by HTML5

As ASP.NET vNext rolls around, it's only natural for you to have questions around the use of Telerik control suites. Let's try to address those.



Question 19:

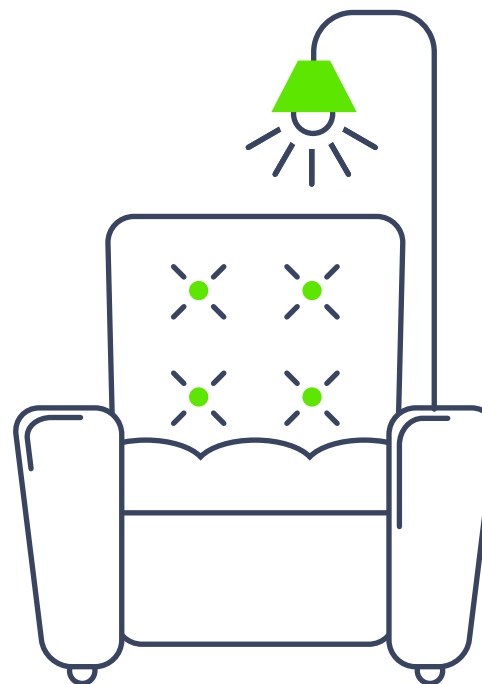
I use Telerik UI for ASP.NET Ajax suite in my ASP.NET WebForms application. Will it work in ASP.NET 5?

What would I have to change?

Answer:

Relax. Like we said before, nobody is moving your cheese. If you use Telerik UI for ASP.NET Ajax in your web applications, it will continue to work as usual in ASP.NET 5 with full WebForms support when targeting the full .NET CLR. Also rest assured, as ASP.NET 5 bits are finalized, Telerik engineers will test our control suites thoroughly for complete compatibility. So nothing has to change from your end, and everything keeps working as usual.

We're excited about the future of ASP.NET web development and are making heavy investments into UI for ASP.NET Ajax product suite. Read all about product usage and roadmaps on our [UI for ASP.NET blog](#).



Question 20:

What else can I expect from Telerik as ASP.NET vNext rolls around?



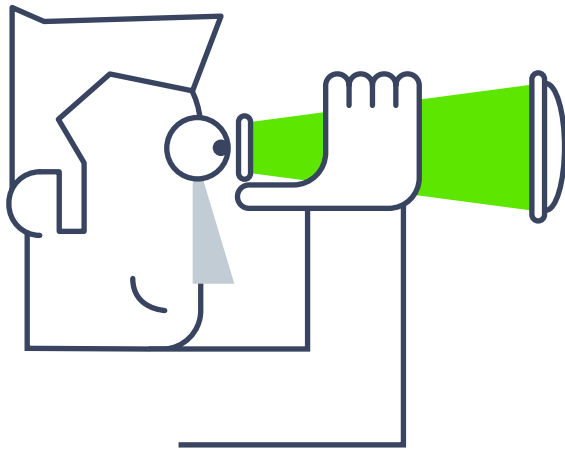
Answer:

In short, plenty of innovations! We're very excited about our [UI for ASP.NET MVC](#) suite, which provides you rich server MVC wrappers to render [HTML5-based modern Kendo UI widgets](#) on the client side. Upcoming enhancements include full support for MVC 6 and .NET Core runtime compatibility.

Bonus Question: :

Fine, I'm sold on ASP.NET vNext.

Where can I find more and what's the release schedule look like?



Answer:

The home for all things ASP.NET vNext is <http://www.asp.net/vnext>.

You should bookmark this to stay on top of regular announcements and community standup meetings. The rough release dates for ASP.NET 5 are Q1 2015 for the first Beta and mid-late summer 2015 for the final release. Exciting times indeed!

Conclusion

I know this was a long read, but I hope I convinced you that the future of ASP.NET development is looking simply awesome. Let's embrace the changes and gear up for a flexible better tomorrow. Best-in-class tooling and an open source cross-platform approach to building the next generation of amazing web apps—that's ASP.NET. Telerik is right here to support your development needs. Now, let's go code!



Sam Basu

Samidip Basu (@samidip) is a technologist, Apress/Pluralsight author, speaker, Microsoft MVP, believer in software craftsmanship, gadget-lover and Developer Advocate for Telerik. With a long developer background, he now spends much of my time advocating modern web/mobile/cloud development platforms on Microsoft/Telerik stacks. His spare times call for travel and culinary adventures with the wife. Find out more at <http://samidipbasu.com>.