



# Future of .NET - .NET on Non-Windows Platforms

A PoV by Aniruddha Chakrabarti AVP Digital, Mphasis

#### Overview

For over a decade now, .NET has been one of the popular platforms for building large-scale distributed applications in enterprise and consumer space. Along with Java/JEE it has dominated enterprise application development. To this day, it has been one of the most popular stacks for building web apps (ASP.NET), mobile apps (Windows Phones, Xamarin), cloud apps (Azure), desktop/smart client apps (Windows Store apps, WPF, and Windows Forms), and apps for Xbox, Kinect etc. Over the years, it has changed itself significantly adding new capabilities and redesigning some of its existing capabilities.

One of the biggest challenges that .NET has faced for even wider adoption is its dependency on Windows. When Microsoft planned .NET in 2000/2001, the world was different – Microsoft purposefully took a dependency on their popular Windows operating system. With the upcoming .NET 5.0 (probable release in 2016), Microsoft has taken a big bet on making .NET open source and cross-platform compatible. Microsoft had released the source code of many components of .NET including ASP.NET and Entity Frameworks some years ago, but with .NET 5.0 Microsoft .NET's team has



completely morphed into an open source community driven culture. The entire codebase is developed openly on GitHub jointly by Microsoft developers and the community.

#### .NET 5.0 would have two flavors:

- A full-fledged .NET Framework 5.0, which would be running only on Windows
- A slim-lined, relatively small, and optimized version of .NET called .NET Core 5.0 which would be running on Linux, OS X, and Windows. .NET Core is a subset of the full .NET Framework and not all features of the full .NET Framework is available on .NET Core



Figure 1 – .NET Framework vs .NET Core in .NET 5.0



Microsoft has changed their strategy significantly over last couple of years towards the Non-Microsoft world. Microsoft understands and acknowledges the fact that within a large enterprise there would be Non-Microsoft software including various Linux flavors as Server OS, Mac laptops and desktops, Android and Apple Phones, Java, Python, Node, and other languages and stacks.

- Azure supports many Linux distros including Ubuntu, CentOS, CoreOS, Oracle Linux, SUSE and latest RHEL
- Azure supports Non-SQL Server databases including Oracle, MongoDB, Redis
- Azure supports running Java, Python, Ruby and Node server side apps. Azure client libraries are available in Java, Python, Ruby and Node, apart from .NET
- Micrsoft Office, for long had a version for Mac. Now Microsoft Office is available for Android and iOS devices as well

The move to create a smaller version of .NET for Non-Windows platforms including Linux and OS X is a major step in Microsoft's move towards a polygot world.

#### .NET story - .NET version 4.6 (till circa 2015)



Figure 2 – Full-fledged .NET Framework on Windows

#### .NET Framework

#### .NET Framework consists of:

#### **CLR (Common Language Runtime)**

CLR or Common Language Runtime is a virtual machine that provides a run-time environment to run the code, and provides services that make the development process easier. CLR is similar to JVM or Java Virtual Machine. Just the way JVM supports multiple languages like Java, Scala, Clojure, etc. CLR supports multiple languages like C#, VB, F#, IronPython (Python implementation running on .NET), etc.

#### BCL (Base Class Library)

BCL or Base Class Library is a runtime library, which provides access to system functionality including IO, Threading, File Access, Data Access, Reflection, Serialization, Networking, Security, LINQ, etc. It is the foundation on which .NET Framework applications, components, and controls are built. It's also called as .NET Framework Class Library or FCL.

### .NET story – .NET 5.0 (circa 2015 onward)



Figure 3 – Different Components of .NET Framework vs .NET Core

#### .NET Core

.NET Core is a cross platform implementation of .NET implemented by Microsoft. .NET Core is a streamlined and relatively small subset of the larger .NET Framework (that was available until now only on Windows). .NET Core is modularized and is available as hundreds of individually deployable NuGet packages. .NET Core packages could be installed from the NuGet repository.

Currently .NET Core is primarily driven so that ASP.NET can run on Linux and OS X apart from Windows. Currently .NET Core supports ASP.NET 5.0, simple Console apps, and Windows 10 UWP (Universal Windows Platform) apps.



Currently .NET Core is primarily driven so that ASP.NET can run on Linux and OS X apart from Windows.

Universal Windows Platform provides a common universal set of runtime APIs for devices running Windows 10 including Windows Store App running on PC and tablets, Windows Phone Apps running on Windows Phone, Xbox Apps, and Surface Apps. This is possible as different Windows 10 flavors running on these difference device types share a common core. With this evolution, apps that target the UWP can call not only the WinRT APIs that are common to all devices, but also APIs (including Win32 and .NET APIs) that are specific to the device family the app is running on. The UWP provides a guaranteed core API layer across devices.

#### .NET Core Libraries are not installed in GAC

(.NET Framework Libraries are installed in GAC) – instead .NET Core Libraries are installed in a single subfolder under the 'Users' folder which means that it does not require admin rights. .NET Core install does not use Windows registry.

There could be multiple versions of DNX or .NET Execution Environments or .NET versions in a machine.



Figure 4a – .NET Core Libraries in "Users" folder

VINDOWS (C:)	• users • a	aniruddha.c 🕨 .dnx 🕨 runtimes 🕨 dnx-clr-win-	x86.1.0.0-rc1-final > b	oin → 🚽 4	Search bin	
le <u>E</u> dit <u>V</u> iew <u>T</u> ools <u>H</u> elp Organize ▼ ■ Open with	Share w	ith 🕶 New folder			# • 🗊	,
🔆 Favorites	-	Name	Date modified	Туре	Size	
E Desktop		Microsoft.Dnx.Compilation.dll	11/16/2015 4:03 PM	Application extens	54 KB	
🐌 Downloads		Microsoft.Dnx.DesignTimeHost.Abstracti	11/16/2015 4:03 PM	Application extens	14 KB	
Secent Places		Microsoft.Dnx.dll	11/16/2015 4:03 PM	Application extens	13 KB	
		Microsoft.Dnx.Host.Clr.dll	11/16/2015 4:03 PM	Application extens	34 KB	
📜 Libraries	=	Microsoft.Dnx.Host.dll	11/16/2015 4:03 PM	Application extens	56 KB	
Documents		Microsoft.Dnx.Host.Mono.dll	11/16/2015 4:03 PM	Application extens	21 KB	
J Music		Microsoft.Dnx.Loader.dll	11/16/2015 4:03 PM	Application extens	18 KB	
E Pictures		Microsoft.Dnx.Runtime.dll	11/16/2015 4:03 PM	Application extens	284 KB	
Videos		Microsoft.Dnx.Runtime.Internals.dll	11/16/2015 4:03 PM	Application extens	13 KB	
		Microsoft.Extensions.PlatformAbstractio	11/16/2015 4:03 PM	Application extens	23 KB	
Normal Computer		System.Collections.Immutable.dll	11/16/2015 11:39	Application extens	196 KB	
awindows (C:)		System.Reflection.Metadata.dll	11/16/2015 11:39	Application extens	281 KB	
👝 DATA (D:)	-	svcruntime140.dll	11/16/2015 11:39	Application extens	78 KB	

Figure 4b – .NET Core Libraries in "Users" folder.

### .NET Core consists of:

#### Core CLR

\* =

Core CLR is a subset of the larger CLR or Common Language Runtime. It is cross-platform, with multiple OS and CPU ports. It includes the garbage collector, JIT compiler, base .NET data types, and many low-level classes.

#### Core FX

Core FX is a stripped down subset of larger BCL or Base Class Libraries for .NET Core. It includes classes for collections, file systems, console, XML, async, and many other components of BCL. Since Core FX is a slimmed optimized version of full BCL that could be ported to Linux and OS X, all APIs of BCL are not present in Core FX.

All of the .NET Core libraries are distributed as NuGet packages. These NuGet Packages could be installed easily within Visual Studio or with one of the NuGet clients directly.



Figure 4 – CLR & BCL vs. Core CLR & Core FX in .NET 5.0

### Next Gen JIT ("RyuJIT")

RyuJIT is Microsoft's next gen 64 bit just in time (JIT) compiler for .NET. According to the **performance analysis** done by .Net runtime team, the new next-generation X64 RyuJIT compiler is twice as fast compared to the older X64 compiler. This means apps compiled with RyuJIT will deliver up to 30% faster start up. RyuJIT was released by Microsoft along with **.NET 4.6 release** and it forms a core part of .NET 5.0 as well.

### .NET Compiler Service ("Roslyn")

**Roslyn** is an open source compiler for C# and VB with rich code analysis APIs. Microsoft developed Roslyn from ground up and redesigned both C# and VB compilers to use services provided by Roslyn. Typically, older generation compilers are complete black boxes, and IDEs or other tools cannot use them as services. Roslyn exposes its functionality as services, which could be easily consumed by IDEs and other tools (code analysis tools for example).

### **LLILC Compiler**

**LLILC** (pronounced as 'lilac') is an LLVM based MSIL Compiler for .NET Core. It includes a set of cross-platform .NET code generation tools that enables compilation of MSIL byte code to LLVM supported platforms.

**LLVM** is a very popular open source compiler platform. LLVM was originally implemented for C and C++. Later it gained widespread popularity and used by many language compilers including Common Lisp, Ada, D, Fortran, Go, Haskell, Java bytecode, Julia, Objective-C, Swift, Python, R, Ruby, Rust, Scala, and Lua.

LLILC creates a bridge into LLVM for .NET, making LLVM's broad chip support and tools available to .NET Core.



.NET Core is a **.NET Foundation** project. Microsoft created .NET Foundation in 2014 to foster open development and collaboration around the growing collection of open source projects for .NET – it's an independent forum driven by the community and Microsoft.

.NET Foundation owns and drives many other open source .NET projects apart from .NET Core including ASP.NET 5, .NET Compiler Platform ("Roslyn"), Entity Framework, ASP.NET MVC, Web API, ASP.NET SignalR, WCF, MSBuild and NuGet. Projects owned by .NET Foundation are open source and licensed under an open source license model.

### ASP.NET 5.0

**ASP.NET 5.0** is the cross platform version of ASP.NET that runs on Windows, Linux, OS X, and any other .NET Core platforms. It's the latest version of several ASP.NET technologies, including ASP.NET MVC and Web API (Web Forms are not supported in ASP.NET 5.0)

Until 4.6, ASP.NET was supported only on Windows. Microsoft took the dependency on Windows long back when ASP.NET was first designed back in 2000-01.

Now with the changed scenario, Microsoft wanted to enable running ASP.NET on Non-Windows platforms including Linux distros and OS X with ASP.NET 5, an ASP.NET website/webapp could be deployed on Linux and OS X servers that would open up many scenarios.

### New .NET 5.0 Command Line Tools

**DNVM:** DNVM or .NET Version Manager helps in installing and managing the versions of .NET runtimes or DNX (.NET Execution Environments) installed on a machine.

DNVM is similar to utilities like NVM (Node Version Manager) and RVM (Ruby Version Manager) that helps in managing and installing versions of Node and Ruby respectively.

C\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\aniruddha.c>dnvm
/#J         /.vcl-rcl-15526           by Viccosoft Open Technologies, Inc.         .vcl-rcl-15526           tigge: One Control (control (contro) (contro) (control (contro) (control (control (control (contro)
Current Fad settings: Defails table: https://www.nuget.org/pj/v2 Defail twrtable: https://www.nuget.org/F/aspnet/next/api/v2 Current table Override: onces Current Unstable Override: onces
to use override reeds, set DAC_FEED and DAC_ONSTAULE_FEED environment keys respectively
comands:
allas Lists and manages allases exec Executes the specified command in a sub-shell where the PATH has been augmented to include the specified DNX
help Displays a list of commands, and help for specific commands install Installs a version of the runtime
list Lists available runtimes
run Locates the drx.exe for the specified version or alias and executes it, providing the remaining arguments to drx.exe Installs the version manager into your User profile directory
uninstall Uninstalls a version of the runtime
update-self Updates DWM to the latest version of the runtime and reassigns the specified alias to point at it use Adds a runtime to the PATH environment variable for your current shell Displays the DWM version Displays the DWM version
C:\Users\aniruddha.c>

Figure 5 – DNVM Command Line Utility

DNVM could be used to list all the .NET versions or DNX installed (use DNVM list command). As shown below, there could be multiple versions of .NET runtime installed with one of them being the default.

ers\aniruddha.c>di	nvm list							
e Version 1.0.0-beta7 1.0.0-beta8 1.0.0-rc2-16128	Runtime coreclr clr coreclr	Architecture x64 <mark>x86</mark> x64	OperatingSystem win win win	Alias  default				

Figure 6 – DNVM list showing versions of DNX / .NET runtimes installed

To change to a different version -

C:\Users\aniruddha.c>dnvm use -r coreclr -arch x64 1.0.0-rc2-16128

Adding C:\Users\aniruddha.c\.dnx\runtimes\dnxcoreclr-win-x64.1.0.0-rc2-16128\bin to process PATH



Figure 7 – Changing the default DNX version

To install the latest version of .NET Core use

D:\Work\Play\dotNetCore>dnvm upgrade -r coreclr

To install the latest version of full .NET Framework use

D:\Work\Play\dotNetCore>dnvm upgrade -r clr

DNVM installed the newly released (November 2015) .NET 5.0 RC1 and made it the default.

DNX: DNX or .NET Execution Environment software development kit (SDK) and runtime environment, which contains everything required to build and run .NET applications for Windows, Mac, and Linux.

	NDOWS\system32\cmd.	exe				× -
D:\Work	k\Play\dotNetCore	e>dnvm li				^
Active	Version	Runtime	Architecture	OperatingSystem	Alias	
•	1.0.0-beta8 1.0.0-rc1-final 1.0.0-rc2-16128	clr coreclr coreclr	x86 x64 x64	win win win	default	
						-

Figure 7a – Changing the default DNX version

DNU: DNU or .NET Development Utilities provides a variety of utility functions to assist with development in .NET Core and ASP.NET 5. Most commonly, DNU is used to install and manage library packages in a .NET Core app. It's also used to package and publish a .NET Core application. DNU uses NuGet behind the scenes for package management and deployment.

#### New .NET 5.0 Project System

.NET 5.0 and ASP.NET 5.0 introduces a new lightweight project system. Until .NET 4.6, .NET used **MSBuild** based project system. MSBuild uses XML files for defining projects – for example C# projects uses .csproj files which are XML files that define a project.



Figure 8 – project.json file structure used by .NET Core projects

.NET 5.0 introduces a more agile JSON based project system – project files are actually JSON files (project.json) that describes the project, .NET Framework version that it uses, the dependencies etc. project. json structure is similar to the project files used by NPM (Node Package Manager), Grunt or Gulp.

#### New .NET 5.0 UI Tools

#### Visual Studio Code



Microsoft created a lightweight cross platform editor that supports Windows, Linux, and OS X. On Windows, developers could use the Visual Studio Community Edition that is the free version of the Visual Studio IDE. But on Linux and OS X, developers could use the Visual Studio Code which is free. Apart from Visual

Studio Code, other editors like Sublime, Atom, Bracket or Vim could be used.

<u>Ele</u>	nt <u>V</u> iew <u>G</u> oto <u>H</u> elp				
n					
<u> </u>	✓ WORKING FILES	1 2 "wersion"			Find A₂ R ≠ ← → E X
ې ا	HelloWorld.cs project.json DOTNETCORE Juscode	3 "descript 4 "dependen 5 ), 6 "framewor	Add Selection To Next Find Match Change All Occurrences Change End of Line Sequence Change File Encoding	Ctrl+D Ctrl+F2	
*	HelloWorld.cs project.json project.lock.json	7 "dnx4 8 "dnxc 9 - 10 11 12	change Language Mode         Change           change Mindow         Change           change         Change	Ctrl+K M Escape Ctrl+Shift+W Ctrl+C, Ctrl-Insert	<pre>[] args){ orld from .NET Core and Core CLR!"); )); name: ");</pre>
		13 } 14 } 15 } 16 17 18		<pre>itiging new = #estimogy itiging the initial (new) itiging the initial (new) itiging the initial (new) itiging the initial (new) itiging the initial (new) ifight (new) ifight (new) initial (new)</pre>	); Listentringo ("Allane","Blord","Cassal"); en) )) Her bullder - now ImatableListerring.Aulio 

Figure 9 – Visual Studio Code

#### OmniSharp



OmniSharp is a community driven open source project created to improve the development experience of .NET 5.0 especially on Non-Windows platforms and in third-party editors like Sublime Text. It's a set of tools, editor integrations and libraries for developing in .NET. OmniSharp works with a number of lightweight editors including Sublime Text,

Atom, Brackets, Emacs, and Vim. Microsoft's new Visual Studio Code editor also uses OmniSharp.

#### Generator-ASP.NET



Yeoman is a scaffolding platform built on top of Node.js that allows you to build template-based generators for projects or code files. generator-aspnet is a yeoman generator that allows you to scaffold ASP.NET 5 applications.

### <br/> <br/>

. . .

The future of .NET lies in successfully running it on Non-Windows platforms, modularizing it fully and developing it openly along with the community.

\*

### Conclusion

The future of .NET lies in successfully running it on Non-Windows platforms, modularizing it fully and developing it openly along with the community. Currently .NET Core only enables ASP.NET 5.0 apps, simple console apps and Windows 10 WUP apps, but we believe that in the future, Microsoft would enable other workloads. It would be not be surprising if .NET components like WCF (Windows Communication Foundation) and WF (Workflow Foundation) are the next set of things to be ported on to non-Windows platforms. Being able to run .NET on Non-Windows platforms would open up many possibilities for Microsoft and large enterprises who have made significant investment for .NET.

Note: As the time this POV was published, Microsoft announced that .NET Core 5.0 would be called .NET Core 1.0 and ASP.NET Core 5.0 would be called ASP.NET Core 1.0. The primary reason behind this decision is .NET Core and ASP.NET Core is not backward compatible and introduces many new ideas. .NET Core is not compatible with .NET 4.6 and hence it's better to call it as .NET Core 1.0 so that developers and architects do not confuse .NET Core with full-fledged .NET 4.6 Framework.

Copyright ©: Third party product and brand names mentioned in this POV (.NET, Microsoft, OS X, Apple, Linux, Xamarin) belong to their respective owners.

### **Further Reading**

.NET Framework	https://msdn.microsoft.com/en-us/vstudio/aa496123
• CLR	https://msdn.microsoft.com/en-us/library/8bs2ecf4(v=vs.110).aspx
• BCL	https://msdn.microsoft.com/en-us/library/gg145045(v=vs.110).aspx
.NET Core	https://dotnet.github.io/
Core CLR	https://github.com/dotnet/coreclr
Core FX	https://github.com/dotnet/corefx
.NET Core Documentation	http://dotnet.readthedocs.org/en/latest/
.NET 4.6 Release	http://blogs.msdn.com/b/dotnet/archive/2015/07/20/announcing-net-framework-4-6.aspx
• Roslyn	https://github.com/dotnet/roslyn
• ASP.NET	http://asp.net/vnext
• ASP.NET 5.0 RC (Release Candidate)	https://get.asp.net/
.NET Foundation	http://www.dotnetfoundation.org/
.NET Core 1.0 and     ASPNET Core 1.0 Announcement	http://www.banselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10ApdNETCore10.aspx



## Aniruddha Chakrabarti

Associate Vice President, Digital, Mphasis

Aniruddha has 16+ years of IT experience spread across systems integration, technology consulting, IT outsourcing and product development. He has extensive experience of delivery leadership, solution architecture, presales, technology architecture and program management of large scale distributed systems.

As AVP, Digital in Mphasis Aniruddha is responsible for Presales, Solutions, RFP/RFI and Capability Development of Digital Practice. Before as Sr. Manager & Sr. Principal Architect in Accenture, he led architecture and large delivery teams. He had played delivery leadership and architecture focused roles in Microsoft, Target, Misys and Cognizant.

His interests include digital, cloud, mobility, IoT, distributed systems, web, open source, .NET, Java, programming languages and NoSQL. His industry experience spans Retail, Healthcare, Capital Markets, Insurance, Travel, Hospitality, Pharma and Medical Technology.

### **About Mphasis**

Mphasis is a global Technology Services and Solutions company specializing in the areas of Digital and Governance, Risk & Compliance. Our solution focus and superior human capital propels our partnership with large enterprise customers in their Digital Transformation journeys and with global financial institutions in the conception and execution of their Governance, Risk and Compliance Strategies. We focus on next generation technologies for differentiated solutions delivering optimized operations for clients.

#### For more information, contact: marketinginfo@mphasis.com

#### USA

460 Park Avenue South Suite #1101 New York, NY 10016, USA Tel.: +1 212 686 6655 Fax: +1 212 683 1690 UK 88 Wood Street London EC2V 7RS, UK Tel.: +44 20 8528 1000 Fax: +44 20 8528 1001

#### INDIA

Bagmane World Technology Center Marathahalli Ring Road DoddanakundhiVillage, Mahadevapura Bangalore 560 048, India Tel.: +91 80 3352 5000 Fax: +91 80 6695 9942



www.mphasis.com