Is .NET a write once, run anywhere (WORA) platform like Java claims to be?

Asked 16 years, 3 months ago Modified 2 years, 9 months ago

Viewed 6k times

▲

10

▼

I remember Sun's slogan so vividly... "Write Once, Run Anywhere". The idea being that since programs are compiled into standard byte codes, any device with a Java Virtual Machine could run it. Over the years, Java seems to have made it onto many platforms/devices.

Is this the intention or was it ever the intention of .NET. If so, what kind of efforts are being put forth to make this a reality?

java .net multiplatform

Share

Improve this question

Follow

edited Jun 23, 2009 at 3:55

community wiki

3 revs, 2 users 100%

Scott Saad

30 Answers

Sorted by: Highest score (default)

8

To correct some comments by others here, .Net was ALWAYS intended to be multi-platform. That is why Microsoft separated the namespaces into "System.*" (which were platform-neutral) and "Microsoft.*" (which were Windows specific).

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:56

James Curran
103k ● 37 ● 185 ● 262

1 Yeah, so effectively using anything from the `Microsoft.*` namespace is bad style because it ruins any hope of portability. – Svante Nov 12, 2008 at 13:50

6

There is Mono which runs on Linux, Solaris and OS X. In practice .Net is still pretty much a Windows-only platform. It's not really in Microsoft's interests to push it to be WORA, on the contrary. Appearing to be cross-platform however is. A lot of people have been really paranoid about Mono on Linux. MS's supposed strategy is to first let it grow to be an important part of the Linux application platform and then release the lawyers. I wouldn't bet my future on .Net's portability.

Share Improve this answer

Follow

edited Sep 17, 2008 at 16:50

answered Sep 17, 2008 at 14:53

auramo

13.3k ● 13 ● 67 ● 88

2 The real strategy seems to be that a lot of windows-only extensions are encouraged so that any open implementation of .net can only provide a subset of the "really used" complete environment, so that every .net program is really a covert windows ad by being a pain to use on other platforms. – Svante Nov 12, 2008 at 13:48

1 Not to mention that Mono is always a release or more behind. Sure, the current release has LINQ, but it's still missing most of the 3.5 features. – cdmckay Jun 23, 2009 at 3:58

4 This is just fear-mongering. Microsoft has placed all versions of the ECMA C# spec and CLI spec under their community promise to not enforce patents on complete implementations of said specs. microsoft.com/interop/cp/default.mspx – Ron Warholic May 24, 2010 at 14:23

@Ron Warholic: Which means that they'll keep their hands off perfect C# and CLI implementations (what about incomplete implementations?). However, the .NET framework is a lot more than that, and I don't think there's any promise on the other stuff. Implementing C# and CLI doesn't do much for cross-platform without the framework. – David Thornley May 24, 2010 at 14:38

1 @David: Mono is a conforming implementation so there aren't issues with it being incomplete. ECMA-335 covers most of the useful libraries (most of the `System` namespace minus WinForms) and there are open-source alternatives to everything not covered (Gtk# etc). If you're 100% paranoid that Microsoft will try to screw you just write code using the open-source components and you'll have no problems. In the context of the question it may not be fully WORA but considering the Mono compiler was built separately from

scratch you can't expect much better. – Ron Warholic May 24, 2010 at 15:34

5

The answer is a very shaky Yes. The moment you include an external library, the answer changes to No.

For example, Microsoft has no 64-bit JET driver. JET is used by .NET to access MS Access databases.

Any applications compiled for the Any CPU target that use MS Access databases will fail on a 64-bit version of Windows.

(This is ignoring that said applications are not portable to Mono.)

Share Improve this answer Follow

answered May 24, 2010 at 14:19

community wiki
Powerlord

4

Microsoft has never made those claims but they ARE making moves in the WORA arena. Silverlight 2.0 for example will use a subset of the .NET framework and be available on Windows, Linux (through the Moonlight project), MacOS, Windows Mobile, and Nokia handsets.

As others have mentioned, the Mono project has also brought the framework to multiple environments.

Share Improve this answer Follow

answered Sep 17, 2008 at 16:43

Michael Brown

9,143 ● 1 ● 32 ● 37

▲

4

▼

To put this in context, in many people's view Java never delivered on its "Write Once Run Anywhere" promise either.

At best what you got was "Write Once Debug Everywhere" or "Write Once Looks like crap Everywhere"

The successful CLR based applications have all been written using a graphical framework that is native to the target platform.

For example the following highly successful linux applications where written using c# bindings to GTK called GTK# and not using winforms like you would expect:

Banshee - music player like itunes

fspot - photo manager

TomBoy - notes program

GnomeDo - Quick launcher and dock

Equally successful windows .net applications are not written using GTK# (even though it is cross platform) they are written using winforms or WPF.

When google came to make Chrome they didn't try to use a cross platform GUI framework, instead they choose to use native GUI frameworks on each platform. Why? because that way the application, fits properly into it's environment, that way it looks, feels and acts like its native to the operating system its on.

Basically when you try to have write once run anywhere you have to make serious compromises and what you end up with is something that doesn't really work right anywhere.

The industry has largely given up on the lofty goal of write once run anywhere, as a nice idea which didn't work out in practice.

The best approach with mono/.net is to share your lower level binaries and to use a native gui framework on each target platform. GTK# on linux, winforms or WPF on windows, CocoaSharp on Mac. This way your application will look and feel like a native app.

Share Improve this answer Follow

answered Nov 22, 2009 at 6:26

community wiki
trampster

Looks like crap... How does a web server look like crap? It cam spit the htm and images it's been given.. You can't blame java foe that. – mP. May 24, 2010 at 14:06

1 That's assuming you use java only for generating HTML... ignoring Swing or other GUIs – Mark Baker May 24, 2010 at 14:24

@mP. I have Jenkins build server and web server can look like crap. At first I thought that if I can do all I need from console, then nothing can possibly go wrong with Jenkins. But that java server is one of the my most painful experience. Integration with git is amazingly terrible. – Atomosk Apr 25, 2017 at 5:54

3

With Mono we're getting pretty close, and with SilverLight we're allready there.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:53

Paul van Brenk

7,540 ● 2 ● 35 ● 39

Silverlight only runs on Intel Macs and Windows. The large base of PPC Macs (especially home and education) ceased to be supported when Silverlight 1.1 introduced the CLR. – Andy Dent Jan 25, 2009 at 5:05

1 Sorry, Silverlight applications doesent work on Linux. The Moonlight Mono project just doesen't have the resources to make that happen. – Jacob Nordfalk Nov 27, 2011 at 1:56

I don't think the official "intention" of .NET was WORA. I think that you could safely say that .NET was designed so that it would always run on future MS OS's. But there is

3

nothing that precludes .NET from running on other platforms. Mono is an example of an implementation of the .NET runtime for an OS other than Windows.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:53

Johnny Bravado

105 ● 2

2

Yes, this was a goal of .NET although I don't think it had the same emphasis as it did in Java. Currently, the only effor that I know of is the Mono project that is creating a version of the CLI which runs on Linux.

Interestingly enough, Silverlight actually has a slimmed down version of the CLR which can run on both Windows and Mac, which allows the same Silverlight app to run on both platforms unchanged.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:54

Scott Dorman

42.5k ● 12 ● 81 ● 112

Was it really a goal of .NET? If it was I would think Microsoft would be making some effort to make the library cross-platform. However, they are not. Once the mono project was mature I believe they threw some money at it, but it is not as though they created an effort to move the CLR to Linux.
– Doug Moore Sep 17, 2008 at 15:24

1

It's theoretically possible, since the CLR (.Net's "virtual machine") complies with an open standard (the CLI). The question is what other implementations there are of that standard. Mono is another work in progress, but it's the only other one I know of.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:53

chessguy
468 ● 5 ● 15

1

I think that the idea with .NET is that it is a "Write Once, Run Anywhere (that Microsoft chooses)". However, the Mono project is slowly changing the situation.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:53

AlexWilson
1,179 ● 2 ● 9 ● 16

2 Which is similar to Java's "Write Once, Run Anywhere with a compatible JVM that doesn't exhibit totally different performance behaviour and/or deprecated functionality between minor revisions" :-) – endian Nov 19, 2008 at 16:39

1

It will never be supported on as many platforms as Java, IMHO.

The only effort is Mono, not sponsored by Microsoft.

Check here on SO and on the official site

Share Improve this answer

Follow

edited May 23, 2017 at 12:34

Community Bot

1 ● 1

answered Sep 17, 2008 at 14:53

Vinko Vrsalovic

340k ● 55 ● 340 ● 373

Actually, I think Mono is co-sponsored by Microsoft and Novell. The problem with Mono is that it isn't very well organized and they are implementing the current features of the day rather than focusing on gettting a stable .NET 2.0 full release and then adding the new language features. – Scott Dorman Sep 17, 2008 at 14:55

They are also always several steps behind. – Doug Moore Sep 17, 2008 at 15:23

1

In theory, yes. .Net Assemblies are bytecodes, which are converted to native code upon startup, using a JIT ("just-in-Time") compiler.

In practice, there aren't many platforms beyond Windows which have a .Net JIT compiler. There's one for Linux, called MONO.

Don't know about Mac, Sun etc...

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:54

James Curran

103k ● 37 ● 185 ● 262

1

Theoretically, the language is designed to be compiled into bytecode like Java which is interpreted by the Common Language Runtime, a mechanism that also allows several languages (not just C#) to work together and run on the .NET framework.

However, Microsoft has only developed the CLR for Windows. There are other non-MS alternatives being developed, the most prominent being Mono, a CLR implementation or a number of platforms (see the link).

So in theory yes, in practice - we'll see.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:55

Christian P.
4,874 ● 7 ● 55 ● 70

1

Yes and no. Parts of the .NET environment are standards and could be openly adopted.

For example, the runtime (CLR) has a portable version called Mono which is multi platform, open source and is used by (for example) Second Life.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:55

Thomas Bratt
51.6k ● 36 ● 126 ● 142

1

The intention, or at least the pitch, was for this to be the case. The reality is that .NET can't really run on other platforms. The only major exception is Mono, which is an open source project. It's essentially a rewrite of the .NET runtime (the equivalent of the java virtual machine) that works on Linux, Solaris, Mac OS X, Windows, and Unix.

It's been fairly successful, but it's not officially supported.

If you're thinking of getting your monolithic Acme corp employer to adopt .Net and Linux, forget it. Realistically, with .NET, you're on Windows machines, period.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:55

Pete Michaud
1,823 ● 4 ● 22 ● 38

1

Yes, .NET has the Common Language Runtime (CLR) which is the .NET equivalent to the JVM. Microsoft does not support it on as many platforms as Java but with the help of the Mono project it is possible to achive cross platform applications with the usual caveats.

Bear in mind that .NET is more than just the CLR. It is a whole platform.

Share Improve this answer Follow

answered Sep 17, 2008 at 14:57

Simon Collins
86 ● 2

Micorosft does not support the CLR on anything but windows. Why do you think pinvoke is everywhere. Why do you think only the language and the entire libraries were standardised. – mP. May 24, 2010 at 14:03

0

Since .NET is only available (officially) on Windows, then not, it isn't write one, run anywhere. However the Mono team are making a good go at helping spread .NET beyond Windows, but they are always way behind the official stuff.

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:54

David Arno

43.2k ● 16 ● 88 ● 132

0

I don't think that it was the original plan, for Microsoft, to create runtimes for every platform and device, but they encouraged this by using a documented (?) intermediate language.

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:54

Grad van Horck

4,506 ● 1 ● 23 ● 22

1 A language without libraries and platforms is useless. – mP. May 24, 2010 at 14:07

0

Multiplatform was of course in the vision.. right now mono does a good job of implementing the runtime for other os.

Mono

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:54

Quintin Robinson

82.3k ● 14 ● 126 ● 132

0

Short answer -- no, Microsoft only supports MS operating systems (including Windows Mobile) for .NET.

Long answer -- there are public open-source projects to replicate the .NET framework for linux and other OSs, notably Rotor and Mono. They don't support everything, but you can deploy a lot of .NET code, including silverlight.

Share Improve this answer

Follow

answered Sep 17, 2008 at 14:54

Guy Starbuck

21.8k ● 7 ● 54 ● 64

0

That depends on your definition of "Anywhere". There are several flavors of Java virtual machine and of .Net framework. And most of the time you can't just write code for a desktop vm/framework and expect it to run on a mobile phone one.

So. in a sense, even Java is not really pure "Write Once, Run Anywhere".

It is true, however, that Java's VM is currently running on several operating systems while .Net framework runs only on Windows devices.

There is one interesting initiative called "Mono" which offers .Net support on Linux, Solaris, Mac OS X, Windows, and Unix. Read here: [Mono Site]()

Share Improve this answer Follow

answered Sep 17, 2008 at 15:01

LiorE
111 ● 4 ● 9

0

dotNet can be, because of the CLR which is similar in function to the JVM. But i dont believe MS had any intention of it being. [http://www.mono-project.com/Main_Page]() might be useful, but its not a MS product. Btw, much like how the wide spectrum of j2ee containers cloud the WORA concept for j2ee apps, ASP.NET apps running on anything besides IIS wouldnt really work the same across disparate platforms.

Share Improve this answer Follow

answered Sep 17, 2008 at 15:07

Sumit
353 ● 1 ● 5 ● 7

I don't think this was ever really a design goal of .NET - Microsoft has no particular interest in people writing

0

software for non-Windows platforms ....

However, there is The Mono project (http://www.mono-project.com), which is "an open development initiative sponsored by Novell to develop an open source, UNIX version of the .NET development platform."

Share Improve this answer Follow

answered Sep 17, 2008 at 16:37

alasdairg
2,118 ● 12 ● 14

0

It was most assuredly meant to be WORA. It's just MS figured Anywhere and Everywhere would be Windows by now. Who knew Linux and the MacOS would still be around. But judging by all the Macs at the PDC, I guess they were either half right or half wrong!

Share Improve this answer Follow

answered Oct 29, 2008 at 6:05

Sevil Natas

0

If WORA was really an original goal, then I guess we'd see .NET implementations on all the major platforms by now, fully supported by Microsoft. I seem to recall that at the time Sun was shouting WORA from the rooftops, Microsoft's riposte was "Write Any (language) Run on One (platform)" (WARO:-). As somebody else mentioned, I think they've always been firm backers of WORASLAIW (Write Once Run Anywhere So Long As Its Windows)

As you point out, they seem to be changing tack a bit with Silverlight to try and get a piece of the Flash/Flex action now that the battlefield has shifted significantly away from the desktop and towards the browser.

Share Improve this answer Follow

answered Nov 12, 2008 at 13:15

alasdairg

2,118 ● 12 ● 14

▲

0

▼

But it IS multiplataform Win9x/WinNT/Mobile

Share Improve this answer Follow

answered Nov 12, 2008 at 13:25

user32415

466 ● 1 ● 5 ● 6

▲

0

▼

Given responses from others I'm still unclear as to whether it was an actual intention of Microsoft to have .NET be a WORA initiative. The only way to really know I guess is to have **somebody from the Microsoft .NET team chime in** on this.

Since we cannot definitively know the original WORA intentions of .NET we can point to efforts that are attempting to make this a reality (as previous answers have talked about).

[Mono] (http://en.wikipedia.org/wiki/Mono_(software))

This effort is an initiative happening outside of Microsoft.

> Mono is a project led by Novell (formerly by Ximian) to create an Ecma standard compliant .NET compatible set of tools, including among others a C# compiler and a Common Language Runtime. Mono can be run on Linux, BSD, UNIX, Mac OS X, Solaris and Windows operating systems.

Silverlight

This effort is being heavily pursued by Microsoft. Silverlight 2.0 implements a version of the framework that is the same as .NET 3.0 and seems to be an attempt to successfully deliver the framework to multiple platforms through the browser.

> It is compatible with multiple web browser products used on Microsoft Windows and Mac OS X operating systems. Mobile devices, starting with Windows Mobile 6 and Symbian (Series 60) phones, will also be supported.

While it does not specifically address bring functionality to GNU/Linux there is apparently a third-party free software implementation named [Moonlight](http://en.wikipedia.org/wiki/Moonlight_(runtime)).

This seems to be what we currently know, but as stated earlier, **it would be very helpful if somebody from the .NET team could pitch in on this one to properly clarify if WORA was in fact an original initiative.**

Share Improve this answer Follow

edited Nov 19, 2008 at 16:20

answered Oct 3, 2008 at 23:05

Scott Saad
18.3k ● 11 ● 66 ● 85

As I just commented on an earlier answer, Silverlight is only now supported on Intel Macs. – Andy Dent Jan 25, 2009 at 5:06

▲
0
▼

If Microsoft were serious about dotnet on other non windows platforms they would have released the class libraries for reuse by others ajoiding the need to rewrite the same libs again. Sun on the other hand has done this meaning less barriers are present ifnone wishesto port to another platform. Natually with java one still needs to write a vm and do the native stuff but it helps avoid a headache that is reimplementing the entire class library. The standardization of the language wS a marketing ploy to grab jon technical folk. A language without libs is worthless. Try doing your next project withnknly the prjkitive types ... That's right write your own string class etc and tell me how helpful a standardiSe language is without any libs available...

Share Improve this answer Follow

answered May 24, 2010 at 14:13

community wiki
mP.

▲

0

▼

I think the idea was to create inter-operability between the different programming languages, not WORA.

Share Improve this answer Follow

answered Apr 28, 2012 at 7:59

community wiki
Nishant George Agrwal

But the idea is the same along the line of WORA. Just like SUN's slogan, you can use .NET framework to program in any Microsoft certified programming languages and run it using CLI. Isn't inter-operability mean the same thing as WORA somewhat? In fact, I've read somehwhere that the idea for .NET came from SUN's compile once and run anywhere. – ThN Nov 12, 2013 at 14:17

0

.Net Core makes .Net "almost" Write Once Run Anywhere.

But there are subtle differences -

1. .Net Core is not really .Net
2. With .Net Core, you write once but build multiple times, once for each specific target OS. Whereas in Java binaries are built once and can be run on any supported OS.

> dotnet build --runtime ubuntu.18.04-x64

Share Improve this answer Follow

answered Mar 10, 2022 at 1:57

community wiki
Nitin Pandey