Is WPF the future of user interface design? Should I learn it now?

Asked 15 years, 8 months ago Modified 14 years ago Viewed 10k times

5

There has been a lot of talk surrounding the likes of WPF. I am wondering if WPF will become a new standard for graphical interactive user interface design. Is this where we are headed in terms of windows interfaces? Will it really take off like everyone says it will?

See also

> [Learning Windows Forms vs. Windows Presentation Foundation]
>
> (Contains links to many other useful posts on WPF).

wpf windows user-interface

Share

Improve this question

Follow

edited Jun 20, 2020 at 9:12

Community Bot
1 ● 1

asked Apr 2, 2009 at 13:18

kevin bailey

This is not a dupe. It may not be a great question but this is certainly a different question than the one referenced.
– JaredPar Apr 2, 2009 at 13:27

@JaredPar - very well. Edited. – Jon B Apr 2, 2009 at 13:30

10 Answers

Sorted by: Highest score (default)

▲
7
▼

I think there are plenty of applications still done in Win32, MFC and of course, WinForms. I think it would be a wise choice to add WPF to your tool belt. Should you drop everything and learn it today? That's up to you. I am seeing more demand for WPF. It's not overwhelming, but neither was C#/WinForms in 2001.

So the long winded answer is that you just have to take the chance. No one knows if WPF apps will dominate the market. I'm leaning towards the possibility and I'm also thinking Silverlight may be a real player in web apps moving forward. Since there are transferrable skills between the two, I'm hedging my bet a little bit by continuing to learn WPF.

Please see also Is it better to use WPF over WinForms

Sorry it's not a concrete answer.

Share Improve this answer

Follow

edited May 23, 2017 at 11:55

Community Bot

1 ● 1

answered Apr 2, 2009 at 13:25

billb

3,619 ● 1 ● 34 ● 36

6

You're asking us to predict the future :)

I think a better way to approach this is to look at the other technology you could learn if you didn't learn WPF. I would weigh the various tradeoffs and pick the one that was more valuable to me.

For instance if the choice was WinForms or WPF I would certainly go with WPF. WPF has a steeper learning curve than WinForms. However once you get past that learning curvie it is so much easier to work with. WPF can do in a few lines what took several hundred lines of a custom control in WinForms.

Share Improve this answer

Follow

answered Apr 2, 2009 at 13:26

JaredPar

753k ● 151 ● 1.3k ● 1.5k

4

WPF is an **ultimate** graphic platform for Windows. Win32's GDI was a "first try", WPF is a "permanent structure". For the combination of Windows and flat displays (f.e. 3d displays might require something else), it will never be replaced. So learn it, it is a good commodity.

Share Improve this answer

Follow

answered Apr 2, 2009 at 20:52

Sergey Aldoukhov

22.7k ● 18 ● 74 ● 104

4

There is hell lot to learn in WPF. You need to die and reborn as a GUI programmer.

But is it worth the effort. Why?, Here is my answer.

Since you are asking this question, I assume you are Microsoft technologies based programmer.

As the direction of MS is towards WPF for GUI development, I see no choice. Win Forms will last long for probably 2 years more. Since the cool look and feel of WPF make users to ask for more and more WPF applications than Win Forms. As you know for many users GUI is the S/W :)

Now if you are non MS based programmer, probably from Java, I say WPF has lot of similarities with Java Swing. But it is a very-very big super set of Swing.

To have Swing catchup with WPF might take at least 2/3 years and by that time WPF might be ruling the word and

I don't expect Swing to be much easier than this, if not difficult.

As silverlight is kind of platform independent and as it's model is similar to WPF, I predict WPF is going to rule at least for next 6/7 years if not a decade.

I believe and hope MS would make things much more easier for the programmers so that learning curve would be shortened or delegated to GUI artiists (using expression blend).

Hope I answered your question.

Share Improve this answer Follow

answered Dec 6, 2009 at 20:48

Satya Dev Bulusu

41 ● 1

3

Microsoft has a habit of throwing everything in the wall and seeing what sticks... The Pocket PC platform, J#, and so on. With regards to WPF, it is too early to tell if adoption will increase in the future.

If you have programmed .NET Winforms and/or Webforms, the learning curve is not that steep. I would suggest dabble with it but don't throw all the eggs in the proverbial WPF (or even Silverlight) basket. As the others have noted, better to treat it as just another tool in your arsenal.

Share Improve this answer

answered Apr 4, 2009 at 2:19

Follow

cyclo
371 ● 2 ● 6

2

WPF has been around for a few years now and Microsoft's decision to rewrite Visual Studio (2010) in WPF is a good sign that it is here to stay. Remember, this is one of the most popular IDEs on the market and a sign of intent from the guys at Microsoft.

My organization adopted the technology last year and while it has a steep learning curve - you really have to learn to think in different terms - it has paid dividends in the richness of applications we are able to develop. I love winforms and am a big fan of asp.net but what blows me away about WPF is that you are provided with the building blocks and the possibilities are endlesss...

If I were you I would learn WPF for the experience and reap the rewards later. Don't forget - you'll also be learning the core of Silverlight if you adopt WPF - these are two technologies that in my humble opinion are going nowhere!

Share Improve this answer

Follow

answered Apr 4, 2009 at 2:25

Jason Irwin
2,045 ● 2 ● 29 ● 42

2

Using WPF is way better then WinForms and you need to have different mindset. All I can say is Microsoft should have used HTML syntax when creating WPF and

Silverlight applications so that front end coulde reused or at least for silverlight apps so that people that develope on Desktop could reuse the same code when writing browser apps that could be used anywhere. If HTML5 becomes better I'm sure it will become popular as trend is toward open source (cheap technologies). No doubt WPF is far better for developing desktop apps then anything else I've used and c# is more powerfull as language (not speed) and how it's used.

Share Improve this answer Follow

answered Dec 8, 2010 at 12:35

Chedo

21 ● 1

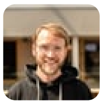
1

Yes start learning it. It's applicable to Silverlight (though not a 1 to 1 mapping), it's also a very similar model to Abobe Flex's paradigm of MXML So you'll be getting 3 wins for the price of 1.

We're starting to see work come in that calls for it, so there's definitely a good reason to have it on the old utility belt.

Share Improve this answer Follow

answered Apr 2, 2009 at 13:32



Rob Stevenson-Leggett

35.7k ● 21 ● 91 ● 142

I am begining to learn it Matthew MacDonald has writen a super book about it. I recommend that book to everyone (Infact I was surfing internet to learn WPF till I came

1

across with his book and one more thing "stay away from Microsoft site (MSDN)"

Share Improve this answer

Follow

answered Dec 20, 2010 at 3:53

Mehmet

41 ● 1

0

Yes, if you will be designing desktop applications on the Windows platform, WPF is the emerging standard. WPF replaces the Win32 API that has dominated the Windows desktop until now, and Microsoft expects a similar lifetime for the WPF platform.

Besides, it's way cooler.

And then there is Silverlight, of course.

Share Improve this answer

Follow

answered Apr 2, 2009 at 13:27

Tor Haugen

19.6k ● 9 ● 47 ● 64

WPF doesn't replace Win32, it is built at a higher abstraction layer that Win32. – Ashley Davis Apr 20, 2009 at 15:42

Yes it does, it is independent of Win32 (even though it interoperates pretty well with it), not an abstraction of it. – Tor Haugen Apr 28, 2009 at 12:58

How can anything on a Windows OS be independent of Win32? I thought Win32 is the basic API for everything that the OS does. – mooware Dec 6, 2009 at 20:54

3 What I mean is the GUI is independent of Win32. I'm referring to those parts of the windows API traditionally used

by applications to interact with the user: GDI (gdi.dll), GUI (user.exe), common dialogs (comdlg32.dll), common controls (commctrl.dll). This is the bulk of Win32 used by traditional apps, and no longer in use by WPF. Instead, WPF builds on DirectX to build the user interface. – Tor Haugen Dec 8, 2009 at 9:21