

Is Learning the win32 API Worthwhile? [closed]

Asked 15 years, 10 months ago Modified 5 years, 3 months ago

Viewed 15k times



38



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 12 years ago.

I was certain that somebody would have specifically asked this question, but from what I can see no-one has (there's been a question about [learning win32](#) but that doesn't cover whether it's worthwhile doing so).

I am very interested in gaining a deeper understanding of all the systems I use (I mostly program in C#, at least professionally), so I wondered, very simply - is learning win32 worthwhile, or is it overkill? Am I wasting my time? Is the knowledge I'd gain worth the effort?

Similar / related questions on StackOverflow:

[Does it still make sense to learn low level WinAPI programming?](#)
[How relevant is Win32 programming to modern professionals?](#)

winapi

Share

Improve this question

Follow

edited May 23, 2017 at 11:54



Community Bot

1 • 1

asked Feb 22, 2009 at 23:13



ljs

37.8k • 36 • 109 • 124

19 Answers

Sorted by:

Highest score (default)



26



Having a working knowledge of how Win32 works at the lowest level will certainly be invaluable if you are planning on doing Windows development in the future. It gives you a level of insight into things like Windows, Messaging and GDI that are hidden by the time you get to the level of .NET.



I wouldn't recommend you try and use Win32 for writing all your applications, but I feel that any Windows developer would benefit from writing a simple Win32 application using C/C++.

This is less true for things like WPF where there is less dependency on Win32, but just knowing how Win32 works will help you understand or appreciate some of the design decisions in WPF.

Share Improve this answer

Follow

edited Sep 8, 2019 at 10:00



Aykhan Hagverdili

29.8k ● 6 ● 48 ● 103

answered Feb 22, 2009 at 23:21



Andrew Grant

58.8k ● 22 ● 131 ● 144

Sure, and of course I'd only be interested in using it where necessary. – [ljs](#) Feb 22, 2009 at 23:55

- 1 @Andrew Grant: I completely agree - since so much of the .NET library is Win32 based, I often find myself understanding their design choices much quicker when I remember that in many cases they're just wrapping the underlying Win32 APIs. – [Mike](#) Feb 24, 2009 at 16:42
-



22



I advocate learning the **concepts behind low level windows programming** if all of the following are true.

1. You are going to do any windows programming.
2. You want to be the "go to" guy when the unexplainable happens.
3. You love to learn.

Abstraction layers like .NET work create and allow developers to do incredible things without having to know

a lot. However, when .NET is used in a way unanticipated by its authors which reveals one of its subtle bugs, then that is the time where some win32 API knowledge goes a long way.

Will you ever have to write a message pump? I doubt it. Can it help diagnose problems? You betcha!

Share Improve this answer

Follow

edited Jun 24, 2017 at 22:16



Cem Kalyoncu

14.6k ● 4 ● 44 ● 62

answered Feb 22, 2009 at 23:36



Glenn

8,022 ● 3 ● 31 ● 39

Indeed. It's really convenient to know your way one (or two) levels down, because [abstractions leak](#). – André Caron Dec 1, 2011 at 21:19

- 1 And here we are, some 6 years later, with the Windows Runtime/the Universal Windows Platform as the designated successor to .NET. Native (i.e. C and C++) wrappers around the Windows API. Does it help to know the Windows API? More than ever! – [Inspectable](#) Jun 15, 2016 at 17:50
-

I loved your answer... – [Vinicius Gonçalves](#) Jul 19, 2017 at 16:52



The question is much like, "Is learning **assembly** worthwhile"; and the answer is the same:



"Yes, because you will understand the fundamentals, and be able to perceive deeper than those who only work at the top level of abstraction".



However, by the same token, you *probably* won't be writing Win32 API directly 99.5% of the time.

Share Improve this answer

answered Feb 22, 2009 at 23:46

Follow



Paul Nathan

40.2k ● 30 ● 120 ● 215



11



When they invented C to replace assembly language, people were probably asking: "is it worthwhile to learn assembly language?" The value in knowing both was being able to drop to assembly to do the things which were impossible to accomplish in C (eg. trigger an interrupt).



The same can be said for Win32. There are some things which are impossible to do in C#. If you didn't know the win32 api, then you would dismiss some things as being impossible. However, once you know what you are missing, in those rare situations, you would be able to "drop to win32" and do them.

Another way of looking at it is this: programming is all about being able to think in multiple levels of abstraction at the same time. For example, if you know your language uses immutable strings, you don't write an algorithm that adds a single character to one 10000 times, because it will be slow. If you know the win32 api,

you will be able to think about how each line you write in C# is actually implemented and that will help you write better code.

Share Improve this answer

answered Feb 22, 2009 at 23:57

Follow



Steve Hanov

11.5k ● 16 ● 66 ● 72

And with the [reference source](#) available, knowing the Windows API is extremely helpful in understanding the implications of certain implementations. You will inevitably make better decisions if you are in the know. – [Inspectable](#)
Jun 15, 2016 at 17:54



8



At least for me, learning an API (I'm assuming that "in-depth" is implied) that I don't use is a waste of time. I'd rather spend my limited amount of time and brain power learning new concepts or exploring new tools than becoming intimately familiar with an existing tool that I don't need to use now. When I need a particular tool that I don't have or have to use a tool that I'm not familiar with, that's the time to learn it in some depth. Before that I might do enough investigation to know whether it is going to be useful to me or not, but not much more.

Share Improve this answer

answered Feb 22, 2009 at 23:23

Follow



tvanfosson

532k ● 102 ● 699 ● 798



7



Yes, you should learn the basics of how Windows (a lot of this stuff predates Win32) operates. Why? For the same reason as I understand how a mortise and tenon joint operates, even though I don't make my own furniture, or why I understand how an internal combustion engine works even though I don't do my own car maintenance.

You work at a higher level of abstraction, which is nice, but when that [abstraction leaks](#) - that's not an "if", that's a "when" - if you don't understand the basics of Windows, you'll be lost. If you don't know at least some of the API, you won't have a clue where to look if you need to P/Invoke functionality not available in .Net.

Quite apart from that, isn't *curiosity* reason enough?

Share Improve this answer

answered Feb 23, 2009 at 11:15

Follow



Bob Moore

6,894 ● 3 ● 32 ● 44



6



Yes, the principles of the Win32 API are useful to learn - these principles are the foundation on which everything else is built.

The .NET APIs for GUI development, both Windows.Forms and WPF, do what they do within the constraints of what is possible on top of the Win32 API. Key architectural decisions of these frameworks were constrained and informed by the Win32 API.

On the other hand, you are less likely to get a lot of value from diving deep into the API, as there is a lot to learn, and given that you spend most of your time working in C#, you'll have less opportunity to use the knowledge directly.

BTW, the same applies to other technologies as well - like networking, cryptography and hardware design. Learning the fundamentals will help you become a better developer.

Share Improve this answer

edited Feb 23, 2009 at 6:13

Follow

answered Feb 22, 2009 at 23:38



Bevan

44.3k ● 10 ● 84 ● 133



4

If you are trying to write a VB6 application then the Win32 API allows you to do a lot of things that are not natively supported by VB6.



If you're writing a C# WinForms app then I would recommend learning the vast reaches of the .NET Framework first.



Edit

If you *really* want to know what's going on under the hood in Windows then you might want to check out a copy of

[Programming Windows 3.1](#) by Charles Petzold.

Share Improve this answer

edited Feb 23, 2009 at 9:10

Follow

answered Feb 22, 2009 at 23:16



Richard Ev

54k ● 61 ● 194 ● 280

I already have a pretty good (well given the huge vastness of the .net framework it's all relative) working knowledge of WinForms, but of course that's a leaky abstraction and I have no real depth of knowledge of the 'under the bonnet' workings of the library... – [ljs](#) Feb 22, 2009 at 23:54

An abstraction - yes, but also known as a more consistent and usable API. Leaky...I hope not! – [Richard Ev](#) Feb 23, 2009 at 9:08



3



I personally think it's still worthwhile learning the Win32 API.

As far as I recall when I started learning Win32 (after doing some VB(A), Pascal, etc.) I learned a lot about Windows and understood how things work in Windows. Everything was so clearer. :)



So, as per your question - you will learn a lot about Windows through learning Win32. As you said - you're a C# programmer and I'm not sure if you'll use it often, because almost everything you need is already there, in .NET.

Share Improve this answer

Follow

edited Jul 13, 2019 at 16:01



Aykhan Hagverdili

29.8k ● 6 ● 48 ● 103

answered Feb 22, 2009 at 23:20



emirc

2,018 ● 1 ● 24 ● 38



2



I won't repeat over and over what the others said many times already.

Here's a link to a [Win32 tutorial](#) with which I am currently learning along the basics of Win32's. I find it pretty interesting and easy to follow.



This tutorial helps me get what I didn't understand first, back when I've begun to program in my secondary school years.

Share Improve this answer

Follow

answered Dec 13, 2010 at 20:58



Will Marcouiller

24.1k ● 26 ● 99 ● 163



1



If I were to start today, I wouldn't learn the entire API. However, I do think that the basic concepts are important to understand, with an understanding of how message loops work as the top priority.

Share Improve this answer

Follow

answered Feb 22, 2009 at 23:43





erikkallen

34.3k ● 14 ● 88 ● 121



1

You'll never be able to just "learn" the entire win32 API, it's too much to take in, and it will be a moving target. If you develop in C#, there's no real point.



That said, try creating Notepad using plain C and just API calls. That will teach you enough for a C# developer to at least appreciate it.



Share Improve this answer

Follow

edited Feb 23, 2009 at 9:16



mghie

32.3k ● 6 ● 92 ● 133

answered Feb 22, 2009 at 23:28



ste

828 ● 2 ● 9 ● 16

of course I don't imagine I could know all of win32, what I mean is to learn a good subset and get a working knowledge going. Your idea is interesting, sounds like a good project :)

– ljs Feb 22, 2009 at 23:37



1

A lot of the "No" answers here seem to focus on learning the actual methods, structs, and what not available in the API. I'd say yes, but focus not on the individual components of the API, but the overall design and the way it functions. It's much easier to troubleshoot even .NET code when you understand what's happening at the core level of the operating system.





Share Improve this answer

answered Feb 25, 2009 at 5:31

Follow



Chris

28k ● 26 ● 130 ● 227



1



This question looks a bit dated, but I'll answer anyway.

Answer: Complicated: yes. Simple: probably not necessary.

It really depends on what you need to do. If you need to use a feature not current supported by .NET, have-at-er. But be careful, most of the coddling the Framework provides Win32 does not, and if you do something incredibly stupid, your machine WILL bluescreen.

I know when .NET first came out, I had have no interest to learn Win32, .NET was here and it was such an improvement. But the sad fact about Windows is this: all new features in Windows are implemented in native code first, period. If you want to use any part of Windows before .NET wraps it, you're either using Win32 from C++ or Win32 from C# or from VB.NET. .NET is a wrapper, for all the stuff in Win32. So if you can't wait, yes, you can Interop into the lower bowels of the OS if you'd like.

Knowing Win32 and probably one day Win64 (whatever they happen to call it) will always be a useful skill. Any whizzbang technology requires underpinnings somewhere.

Share Improve this answer

Follow

answered May 13, 2009 at 17:17



Chris

6,750 ● 8 ● 46 ● 61



1

.NET is implemented using win32 api, anyone wishing to possess deeper understanding of .NET would greatly benefit from having at least marginal knowlege of win32api.



Share Improve this answer

Follow

answered May 19, 2010 at 16:38



LoudNPossiblyWrong

3,893 ● 7 ● 34 ● 45



0

In your career it will be unlikely that you will only be creating greenfield applications where you will have the freedom to choose the technology and programming languages used.



Sooner or later you will have to integrate with old code written in Win32 and C/C++. In that case, knowledge of Win32 will help, especially if you are integrating using PInvoke or C++/CLI.



Share Improve this answer

Follow

edited Feb 25, 2009 at 5:27



m-sharp

17k ● 1 ● 28 ● 26



Misuse the .NET framework and Win32 and your machine will blue screen? Somehow I doubt that.

0



The biggest value to knowing Win32 (or assembly language) is that when something doesn't work as expected and you have to debug it. The more you know about the underlying system, the easier time you will have debugging the problem.



Share Improve this answer

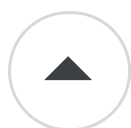
answered Sep 2, 2009 at 23:51

Follow



James

148 ● 1 ● 7



I like to further add to this. I never formally studied Win32 API/MFC. I started using Visual C++ 4 when I first got interested in GUI programming. Anyway, I wish I kept that foundation then, as I never caught on quick enough (I was rather young then, actually), so I studied Visual Basic instead.

0



For some reason, Delphi never interested me even though I knew Pascal well enough, but I digress. These days, I work in IT and develop installer scripts in NSIS - and every so often I need additional functionality that NSIS doesn't provide, so I make my own plugin and to keep it quick and dependency free, I opt for Win32 API opposed to MFC or even full blown C++.

The main reason for this comment, is that my own curiosity got me hooked. I like to know more, so where is

the best resource for learning the API. A book? Website?

Would MFC still be worth tackling as well? I did see a website about a fellow that develops Win32 GUI apps, in assembly! I think that is overkill, honestly, but it is compact, fast code, interesting, the concept, but I never was able to get the hang of 80x86 assembly (hell, even RISC assembly in college I never was able to do!)

Share Improve this answer

answered Sep 24, 2010 at 20:21

Follow



[Jonathan Weinraub](#)

387 ● 5 ● 20

If you're coding for fun,...and assembler is fun to you (God knows why!) then it might be worth writing GUIs this way as a learning exercise but I can't ever imagine telling my boss I was writing something really efficient but it's gonna take me 48 months! – [EightyOne Unite](#) Sep 28, 2010 at 21:48

The tutorial linked in my answer is a good start.
– [Will Marcouiller](#) Dec 18, 2010 at 11:45



0



I think it's always interesting to know how a system works if *your work relies on it*. I don't mean you should learn every bit of it, but still get a good understanding, at least to be able to search more by yourself the day you will have to.



Software is not magic - well... ok... for 99% of the cases :-)



Here is a link to an excellent article about "magical thinking" and "GUID goblins" from Eric Lippert's about that subject: [It's not Magic](#)

Share Improve this answer

Follow

answered Apr 27, 2012 at 13:05



[Simon Mourier](#)

138k ● 22 ● 261 ● 307
