C# Desktop Applications? [closed]

Asked 16 years, 2 months ago Modified 13 years ago Viewed 13k times

4

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 5 years ago.

Improve this question

In looking at the use of C# in a ISV setting, I'm wondering what prominent C# based desktop apps are out there? I can think of only Paint .NET.

Is C# a good idea for an ISV, or should one stick to more native environments like Delphi or even QT?

Of course any experienced based advise or feedback would be appreciated.

c# .net

edited Mar 14, 2009 at 16:51

Share

Improve this question

Follow

BlackWasp
4,961 ● 2 ● 33 ● 42

asked Oct 16, 2008 at 15:34

geo
1,257 ● 4 ● 12 ● 17

22 Answers

Sorted by: Highest score (default)

▲

12

▼

Yes, C# can be used quite effectivly to build applications. In regards what applciations are out there, what are you looking for? Big apps, little apps?

I know of a big app that is built on the .NET framework (unsure if C# or VB) and that is Quickbooks.

Share Improve this answer

Follow

answered Oct 16, 2008 at 15:38

Mitchel Sellers
63.1k ● 15 ● 114 ● 174

4 Quickbooks is pretty well known. – BlackWasp Oct 16, 2008 at 15:51

▲

7

▼

Developing a Desktop applications in C# is great. Its not just for Web Apps.

WinForms are going to save you huge amounts of time. It really is a first class citizen when it comes to desktop windows development. If interop is a problem you can always use P/Invoke and COM object wrapping that VS does for you.

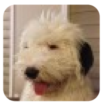
Done right, it will be a breeze to maintain and update when the client changes their mind on what they want.

Yes, the framework needs to be on the machine, But this shouldn't be a problem if they have ever used windows update. Nowadays what language doesn't come with a run time library. In VS you can create simple installers that include the runtime.

Of course if your gonna deploy to Linux and mono, your mileage may very.

Share Improve this answer Follow

answered Oct 16, 2008 at 15:52



discomurray

14.8k ● 1 ● 23 ● 11

5

PlasticSCM

Share Improve this answer Follow

answered Dec 22, 2008 at 13:57

Emrah Diril

1,755 ● 1 ● 20 ● 29

The greatest version control system ever!! try it !! – Mawg
Sep 28, 2013 at 5:32

4

If you can control the system requirements for your application, C# is fine. Some end users still (even though we're nearly in 2009 now) object to a 40MB runtime for some reason, so if you're looking to deploy an application commercially, that may be an issue for you. In a corporate setting, though, where there is some standardization of software on users' computers, this is probably not a problem.

Delphi and QT specifically are both problems. Delphi is effectively a dead language. Companies that are using it these days are, for the most part, porting their code away from it as fast as they can (job boards seem to be full of Delphi-to-C# migration jobs these days). You may like QT, but that moves almost as slowly as Delphi in the Windows world, so I would never consider it to be a real option.

Share Improve this answer Follow

answered Oct 16, 2008 at 15:38
TheSmurf
15.6k ● 3 ● 41 ● 48

Delphi is more popular than Ruby in TIOBE's top 10 index. Are you saying ruby is a dead language also? Delphi is just dead in the US, it's very much alive everywhere else. And 40 MB is big if you host centrally on a file-share.
– Joeri Sebrechts Oct 16, 2008 at 16:06

Why would you assume the only options are "dead" and "not dead?" Ruby's not a dead language; it's a new language (by comparison). My point about 40MB was that if you're in a corporate environment, you have the ability to standardize software ON USER MACHINES, and the size is irrelevant. – TheSmurf Oct 16, 2008 at 16:45

The biggest telecom company in my country uses standardized desktops, but does not include most software on the machines because the configuration follows the profile, not the machine. Our software runs from a file-share on their network. – Joeri Sebrechts Oct 16, 2008 at 20:53

I think that's a little different. .NET is not software that you run; it's something that has to be installed on the machine. Anyway, I WAS talking about consumer dislike of installing a runtime, not questionable corporate setups. – TheSmurf Oct 17, 2008 at 2:23

4

More:

- Business Plan Pro
- Rescue Time client
- Sony Vegas
- Sql Management Studio
- VS 2010 (much more so than VS 200x)
- NASA's World Wind
- BabySmash
- Windows Live Writer
- Microsoft Office Accounting
- Fiddler 2
- Windows Mobile Device Center
- AdiIRC

- Jetbrains dotTrace
- Lots of Lenovo utilities
- Planbook

These are just the ones installed on my PC...

Share Improve this answer Follow

answered Mar 14, 2009 at 17:00

codekaizen
27.4k ● 7 ● 86 ● 140

▲

3

▼

I think that Microsoft Expression Blend/Studio is written in C#

Share Improve this answer Follow

answered Oct 16, 2008 at 15:41

Sean
62.4k ● 11 ● 99 ● 138

Yes it is :) The next version of visual studio is all manged. its new UI is written in WPF – Foredecker Dec 21, 2008 at 16:51

▲

3

▼

Anything done for Robotics Studio, any XNA game (quite a few commerical ones are coming out now)

Share Improve this answer Follow

answered Oct 16, 2008 at 15:53

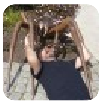
TraumaPony
10.8k ● 13 ● 57 ● 75

3

I don't see why you would not use C# as an ISV. The problem is ensuring that your target market has the .NET framework. If you are using 3.5, you can build a mini version into your distribution though this increases the file sizes for downloads of course.

Share Improve this answer Follow

answered Oct 16, 2008 at 15:53



BlackWasp
4,961 ● 2 ● 33 ● 42

3

Some supplemental information to think about:

One of the typical points against doing desktop applications the distribution model is a lot more difficult to manage. With web apps, if you have a large user base, you can do an update and instantly have your entire user base running on the latest version.

With traditional desktop apps, you'd have to send out an .msi or build something yourself. However, with ClickOnce and the Updater Application Block (http://msdn.microsoft.com/en-us/library/ms978574.aspx), it's so easy to build intelligent updating into your application that it might play very nicely into an ISV plan.

Even though VS 2008/WPF is shrinking the gap, it's typically much faster to develop client applications in the

WinForms space that on the web, so I think it's a very viable approach.

Share Improve this answer

Follow

answered Oct 16, 2008 at 18:20

joshua.ewer

3,964 ● 3 ● 27 ● 35

2

I believe TimeSnapper is written in C#

Share Improve this answer

Follow

answered Oct 16, 2008 at 15:38

harriyott

10.7k ● 10 ● 68 ● 104

2

I have worked for an ISV before that used Delphi and it was excellent for their needs. It still produces great native applications and although dieing it is certainly not dead (yet). Until recently I would have recommened Delphi over c# for desktop applications as I had performance and footprint consderations but as .net can now be considered ubiquitous and that the platform is now maturing my opinoin would probably go with c# (over Delphi). If you need to consider employees, you will find that there are fewer Delphi developers around that c# ones. So you may struggle to entice good devleopers in as you are using that. That said Delphi is looking exciting again with the Delphi prism VS plugin.

Just as an aside did you know that the c# compiler is written in c++? Delphi was written in Delphi from v1.0

Share Improve this answer

Follow

answered Mar 14, 2009 at 16:59

Johnno Nolan

29.6k ● 19 ● 113 ● 160

1

Windows Media Centre is powered by .NET.

Share Improve this answer

Follow

answered Dec 21, 2008 at 16:37

The Alpha Nerd

105 ● 2 ● 10

0

I believe that MS use C# to build many of their apps both Windows and Web. At this time, C# is a great language to go with because of the flexibility of the language/framework.

Share Improve this answer

Follow

answered Oct 16, 2008 at 15:37

JFV

1,783 ● 6 ● 24 ● 38

0

The zune software (v2 and up) is written in c#. Oh, and Sage Timberline Office is written in .net (mostly).

Share Improve this answer

Follow

answered Oct 16, 2008 at 15:45

Darren Kopp

77.6k ● 9 ● 80 ● 93

0

Although it's a developer tool, the excellent open source IDE SharpDevelop was built in C#. I've also worked on another which (despite being semi-commercially-available) I'm not at liberty to divulge. (Very vertical market.)

I'm a huge fan of the C#/WinForms combination, and can't see why you wouldn't go that route if your developers are already familiar with C# and you want to get into the desktop space.

Share Improve this answer

Follow

answered Oct 16, 2008 at 15:59

John Rudy

37.8k ● 14 ● 66 ● 101

0

A very prominent example for a commercialy successful application would be Microsoft Visual Studio 2005 itself. If you are looking into building a client or server application for the Windows ecosystem .Net with C# is an absolut valid choice. You'll get a very rich framework, a great third-party-app ecosystem and a huge community. If you need a minimal memory footprint for your application or are very hardware centric then C++ maybe an option.

If you want to target the UNIX/LINUX or Apple platforms you should be looking in frameworks more native for the respective platform, though you can get a certain degree of interoperability with the Mono project (But I am not up to date on how complete their implementation of .Net currently is).

Share Improve this answer Follow

answered Oct 17, 2008 at 8:40

lowglider
1,087 ● 2 ● 11 ● 21

0

I believe the matrox graphics card configuration utilities are using the .net-framework.

Share Improve this answer Follow

answered Dec 21, 2008 at 16:33

EricSchaefer
26.3k ● 21 ● 74 ● 105

0

I'm still just a student, but for what it's worth (and from other answers preceding mine), there seem to be quite a few apps in C#.

I'd advise working through the book Head First C# (Amazon link). This book will give you a pretty solid idea of what you can do with the language.

=-MDP-=

Share Improve this answer

Follow

answered Dec 21, 2008 at 16:39

Nerf42

191 ● 1 ● 3 ● 12

[Creative Docs .NET](#) is a very nice example of C# application.

0

Share Improve this answer

Follow

answered Dec 21, 2008 at 16:47

mem64k

748 ● 1 ● 13 ● 25

Microsoft's World Wide Telescope software is written in c#

0

Share Improve this answer

Follow

answered Dec 21, 2008 at 16:53

Erick

Who are your target users? Their needs will dictate your choice of language. C# and VB.NET are good general purpose langs, but if you are targeting Mac or *nix, you may want something like C++.

0

Share Improve this answer

edited Feb 10, 2010 at 5:37

Follow

Neil N
25.3k ● 17 ● 86 ● 146

answered Oct 16, 2008 at 15:40

StingyJack
19.4k ● 11 ● 67 ● 126

windows, currently I develop Cocoa apps. – geo Oct 16, 2008 at 15:45

0

If you're looking for a way to get started quickly, I recently published DesktopBootstrap, which is my attempt to factor out the elements you'll need to get started.

Hope that helps!

Share Improve this answer Follow

answered Nov 27, 2011 at 20:31

adam smith
764 ● 8 ● 16