

How can you make a .net windows forms project look fresh?

Asked 16 years, 3 months ago Modified 11 years ago Viewed 36k times



34



I'm working on a visual studio 2005 vb.net windows forms project that's been around for several years. It's full of default textboxes, labels, dropdowns, datagrids, datetime pickers -- all the standard stuff. The end result is a very gray, old-looking project.

What would be the best approach to making this project look fresh and snazzy? I'd rather not rewrite the entire solution with all brand new forms objects, but would that be avoidable?

.net

visual-studio

winforms

user-interface

Share

Improve this question

Follow

asked Aug 29, 2008 at 0:30



Jeffrey

1,667 ● 5 ● 25 ● 41

7 Answers

Sorted by:

Highest score (default)



I was actually just sprucing up a dialog today. A lot of it depends on what kind of application you have, and what

OS it is running on. A couple of these tips will certainly go a long way to jazzing things up.



1. Ensure adequate spacing between controls — don't cram them all together. Space is appealing. You might also try flowing the controls a little differently when you have more space.
2. Put in some new 3D and glossy images. You can put a big yellow exclamation mark on a custom warning dialog. Replace old toolbar buttons with new ones. Two libraries I have used and like are [GlyFX](#) and [IconExperience](#). You can find free ones too. Ideally get a graphic artist to make some custom ones for the specific actions your application does to fill in between the common ones you use (make sure they all go together). That will go a long way to making it look fancy.
3. Try a different font. Tahoma is a good one. Often times the default font is MS Sans Serif. You can do better. Avoid Times New Roman and Comic Sans though. Also avoid large blocks of bold — use it sparingly. Generally you want all your fonts the same, and only use different fonts sparingly to set certain bits of text apart.
4. Add subdued colors to certain controls. This is a tricky one. You always want to use subdued colors, nothing bright or stark usually, but the colors should indicate something, or if you have a grid you can use it to show logical grouping. This is a slippery slope. Be aware that users might change their system

colors, which will change how your colors look.

Ideally give them a few color themes, or the ability to change colors.

5. Instead of thinking eye-candy, think usability. Make the most common course of action obvious. Mark Miller of DevExpress has a great talk on the Science of User Interface Design. I actually have a video of it and might be able to post it online with a little clean-up.
6. Invest in a few good quality 3rd party controls. Replacing all your controls could be a pain, but if you are using the default grids for example, you would really jazz it up with a good grid from [DevExpress](#) or some other component vendor. Be aware that different vendors have different philosophies for how their components are used, so swapping them out can be a bit of a pain. Start small to test the waters, and then try something really complicated before you commit to replacing all of them. The only thing worse than ugly grids is ugly grids mixed with pretty grids. Consistency is golden!
7. You also might look at replacing your old tool bars and menus with a Ribbon Control like Microsoft did in Office 2007. Then everyone will think you are really uptown! Again only replacing key components and UI elements without thinking you need to revamp the whole UI.
8. Of course pay attention to the basics like tab order, etc. Consistency, consistency, consistency.

Some apps lend themselves to full blown skinning, while others don't. Generally you don't want anything flashy that gets used a lot.

Share Improve this answer

Follow

edited Nov 27, 2013 at 10:25



Uwe Keim

40.7k ● 61 ● 185 ● 301

answered Aug 29, 2008 at 1:42



Jim McKeeth

38.7k ● 25 ● 124 ● 199

2 [Link to the talk that Jim is talking about above.](#)

– DjangoBlockchain Nov 9, 2017 at 16:09



9



One other thing to also check is that your controls have the **FlatStyle** property set to **System** instead of Standard.

What this will do is make sure that the app uses the system defaults for radio buttons, standard buttons and the like. This takes all your apps from the flat Win 2000 look and gives them the XP or Vista bling depending on the OS they are running.

Share Improve this answer

Follow

answered Aug 29, 2008 at 22:13



Dillie-O

29.7k ● 14 ● 102 ● 141



This isn't so much an "answer" as an opinion.

7



I tried to jazz up a WinForms project I created back a few years ago by giving the forms a fancy blue gradient background etc, and it looked pretty good on XP. But then on Vista it looked out of place. Taking away any custom painting and reverting the form to "battleship gray" made it look much better IMHO.

I'm seeing a lot of applications (particularly from MS) coming out with custom window chrome etc, and all it does is detract from the nice sense of consistency that Windows gives.

I guess what I'm saying is that you don't need to worry too much about making your application look fashionable. If you keep your colours based on the SystemColors enumeration then Windows can do that for you.

Share Improve this answer

answered Aug 29, 2008 at 0:42

Follow



Matt Hamilton

204k ● 61 ● 392 ● 321



6



I recommend purchasing a good 3rd-party control library - [Infragistics](#) and [DevExpress](#), are just a couple. Most of these libraries give you the ability to drop in new compatible controls on top of your existing ones - for example, you can replace the default EditBox with an enhanced version. They also give you access to some of the snazzy new UIs such as Ribbon, or the Outlook-style navigator people are always wanting.

The reason I specifically recommend using one of these libraries is that they were designed to be relatively easy to use in existing applications, you get support, a community, and all sorts of upgrade paths/options.

The downside: money.

Share Improve this answer

answered Aug 29, 2008 at 1:23

Follow



user2189331



5

What would be the best approach to making this project look fresh and snazzy?



IMHO the best thing you can do is make sure the controls are logically ordered, and have ample spacing between them, and add groupboxes / labels / etc where appropriate.



If you try and change the 'sea of gray' that is the default color scheme, your app will just end up looking crap.

Share Improve this answer

answered Aug 29, 2008 at 1:06

Follow



Orion Edwards

123k ● 66 ● 245 ● 339



3

This depends on how the existing "gray old looking" project is structured in terms of code. For example, is data access code separated from the UI in a Data Access



Layer, is the business logic in a Business Logic Layer? If yes, then cleaning the UI for a snazzy look should be relatively simple.



If everything is all there in the "Button Click" event, then a rewrite is the only way in my humble opinion as otherwise it will just be too time consuming trying to work with the existing code base.

Cheers

Share Improve this answer

answered Aug 29, 2008 at 0:40

Follow



user352

413 ● 1 ● 5 ● 8



3



You can subclass all the default controls and override their appearance. Admittedly, you will have to go thru the entire project and change all references of TextBox to MyTextBox, but all of the default properties and methods will still work. The same cannot be guaranteed if you go with a 3rd party vendor. The other advantage of this approach is you can pick one control at a time and perform an incremental upgrade of the application.



Share Improve this answer

answered Aug 29, 2008 at 0:43

Follow



Jason Z

13.4k ● 15 ● 51 ● 62