

Folders or Projects in a Visual Studio Solution? [closed]

Asked 16 years, 4 months ago Modified 11 years, 11 months ago

Viewed 6k times



32



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 6 years ago.

[Improve this question](#)

When splitting a solution in to logical layers, when is it best to use a separate project over just grouping by a folder?

visual-studio

projects-and-solutions

Share

Improve this question

Follow

edited Sep 10, 2008 at 21:37



[Mark Cidade](#)

99.8k ● 33 ● 229 ● 237

asked Aug 4, 2008 at 22:00



Dan

29.4k ● 44 ● 151 ● 209

7 Answers

Sorted by:

Highest score (default)



By default, **always just create new folder** within the same project

19



- You will get single assembly (without additional ILMerge gymnastic)
- Easier to obfuscate (because you will have less public types and methods, ideally none at all)



Separating your source code into multiple projects makes only sense if you...



- Have some portions of the source code that are part of the project but not deployable by default or at all (unit tests, extra plugins etc.)
- More developers involved and you want to treat their work as consumable black box. (not very recommended)
- If you can clearly separate your project into isolated layers/modules and you want to make sure that they can't cross-consume **internal** members. (also not recommended because you will need to decide which aspect is the most important)

If you think that some portions of your source code could be reusable, still don't create it as a new project. Just wait

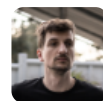
until you will really want to reuse it in another solution and isolate it out of original project as needed. Programming is not a lego, reusing is usually very difficult and often won't happen as planned.

Share Improve this answer

edited Aug 4, 2008 at 22:54

Follow

answered Aug 4, 2008 at 22:45



[lubos hasko](#)

25k ● 10 ● 57 ● 62

1 I like lubos' sense of reality. – [ybakos](#) May 18, 2012 at 15:31

maybe an older answer, but oh man, i wish those consultants working at my company have the point view. I feel that they always complicated things ahead of time like if they wanted to prove their usefulness by spitting out things they will only be able to maintain. – [olleo](#) Jun 9, 2019 at 15:07



8



denny wrote:

I personally feel that if reusable code is split into projects it is simpler to use other places than if it is just in folders.



I really agree with this - if you can reuse it, it should be in a separate project. With that said, it's also very difficult to reuse effectively :)

Here at SO, we've tried to be very simple with three projects:

- MVC Web project (which does a nice job of separating your layers into folders by default)
- Database project for source control of our DB
- Unit tests against MVC models/controllers

I can't speak for everyone, but I'm happy with how simple we've kept it - really speeds the builds along!

[Share](#) [Improve this answer](#)

[Follow](#)

answered Aug 4, 2008 at 22:42



[Jarrod Dixon](#)

15.8k ● 9 ● 61 ● 72



7



Separating features into projects is often a YAGNI architecture optimization. How often have you reused those separate projects, really? If it's not a frequent occurrence, you're complicating your development, build, deployment, and maintenance for theoretical reuse.



I much prefer separating into folders (using appropriate namespaces) and refactoring to separate projects when you've got a real-life reuse use case.

[Share](#) [Improve this answer](#)

[Follow](#)

answered Aug 17, 2008 at 17:35



[Jon Galloway](#)

53.1k ● 25 ● 127 ● 194

1 When I replace 'theoretical reuse' with 'testing', does your conclusion still hold? – [reinierpost](#) Jun 11, 2010 at 14:06



4

I usually do a project for the GUI a project for the business logic a project for data access and a project for unit tests.



But sometimes it is prudent to have separation based upon services (if you are using a service oriented architecture) Such as Authentication, Sales, etc.



I guess the rule of thumb that I work off of is that if you can see it as a component that has a clear separation of concerns then a different project could be prudent. But I would think that folders versus projects could just be a preference or philosophy.

I personally feel that if reusable code is split into projects it is simpler to use other places than if it is just in folders.

Share Improve this answer

answered Aug 4, 2008 at 22:14

Follow



[denny](#)

1,481 ● 1 ● 17 ● 26



1



If you do go for creating several projects, make sure everyone who adds code to the solution is fully aware of the intention of them and do everything you can to get them to understand the dependencies between the projects. If you have ever tried to sort out the mess when someone has gone and added references that shouldn't have been there and got away with it for weeks you will understand this point

Share Improve this answer

answered Sep 10, 2008 at 21:36

Follow



Oskar

2,294 ● 5 ● 28 ● 36



1



I really think it is better to split the project as well, but it all depends on the size of the project and the number of people working on it.

For larger projects, I have a projects for

- data access (models)
- services
- front end
- tests

I got the model from Rob Connery and his storefront application... seems to work really well.

[mvc-storefront](#)

Share Improve this answer

Follow

edited Jan 27, 2013 at 12:12



user1873471

answered Sep 15, 2008 at 23:45



user10479

810 ● 2 ● 11 ● 22



0



Separating your source code into multiple projects makes only sense if you... .. More developers involved and you want to treat their work as consumable black box. (not very recommended) ...

Why isn't this recommended? I've found it a very useful way to manage an application with several devs working on different portions. Makes checkins much easier, mainly by virtually eliminating merges. Very rarely will two devs have to work on the same project at the same time.

Share Improve this answer

Follow

answered Aug 17, 2008 at 17:19



rjrapson

1,997 ● 1 ● 21 ● 28

It's only recommended for code that is sufficiently stable to allow clear responsibilities and interfaces for projects to be determined. – [reinierpost](#) Jun 11, 2010 at 14:12