

What do you do about references when unloading a project in Visual Studio?

Asked 16 years, 2 months ago Modified 16 years, 1 month ago

Viewed 4k times



6



When you unload a project in Visual Studio, any referencing projects get warning triangles on their reference to the unloaded project. I've written myself a macro to do clever stuff (detect add/remove of project and transform any references from-to file/project dependency), but I can't believe that I'm not missing something much simpler. How can the unload function be any use if I have to go around manually changing references (and it breaks the 'personal solutions/shared projects' team development paradigm).

(This question is related to answers to [this question](#) about structuring large solutions in Visual Studio - some answers mentioned having solutions with lots of projects, but 'unloading' unused projects to improve performance.)

visual-studio

reference

projects-and-solutions

Share

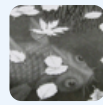
Improve this question

edited May 23, 2017 at 11:49



Community Bot

asked Oct 7, 2008 at 5:55



Benjol

66.4k ● 55 ● 192 ● 275

That macro sounds nice! I've wanted to figure out a macro or addin to do something like that, but haven't gotten around to spending the time to do it. Haven't found anything on the internet. Any chance (if it is reusable enough) you can post it somewhere? – [Mafu Josh](#) Feb 13, 2012 at 14:50 ✎

@MafuJosh, thanks for your interest (and feedback). Here it is (with no warrantee and the usual yadayada):

gist.github.com/1824214 – [Benjol](#) Feb 14, 2012 at 6:24

thank you, I'll look into this when I get a chance (when I'm not working nights and weekends anymore...) and let you know how it goes. – [Mafu Josh](#) Feb 15, 2012 at 19:53

4 Answers

Sorted by:

Highest score (default)



1

For my projects, I create an `assemblies` folder which the projects automatically copy into from a set location to which other projects copy builds.



Post-build for referenced assembly's project:

```
if not exist "C:\builds\Project1" md
"C:\builds\Project1\"
copy "$(TargetDir)$(TargetName).*"
"C:\builds\Project1\"
```



Pre-build for referencing projects:

```
if exist "c:\builds\Project1\" copy  
"c:\builds\Project1*.*)" "$(ProjectDir)assemblies"
```

The project file points to its `assemblies` subfolder for references so even if the source projects are unloaded from the solution, the last-built assemblies will be used without the performance problems of having the whole project in memory while developing.

Share Improve this answer

edited Oct 7, 2008 at 11:12

Follow

answered Oct 7, 2008 at 6:23



[Mark Cidade](#)

99.8k ● 33 ● 229 ● 237

Nice, but does that resolve the problem of references when unloading a project? VS doesn't seem to be smart enough to detect that the dll exists in the reference path. And if you only have file references, there is no advantage of having projects in the same folder. – [Benjol](#) Oct 7, 2008 at 7:30

Even though you don't have direct project references (which aren't useful if you're going to unload them anyway) there are other advantages to having them in the same solution, although you'd only need them in the same *folder* if your SCC provider expects it. – [Mark Cidade](#) Oct 7, 2008 at 11:09



1

What are the advantages of having projects in the same solution if you use file references?



If your `app.exe` uses `utils.dll` and you change the code for `utils.dll`, then if it's in the same solution VS will notice the dependency and recompile both. If it's not in the solution you'll have to jump out, recompile `utils.dll` separately, then jump back in and recompile `app.exe`.



This becomes either more or less important depending on how many other dll's your exe is referencing, and how often they change (in team environments shared dll's change often in my experience).

There is also the side effect that if you have 100 projects in VS it will take a long time to process them all just to figure out if they need recompiling or not.

Share Improve this answer

answered Oct 21, 2008 at 20:19

Follow



Orion Edwards

123k ● 66 ● 245 ● 339

Hi, I moved my 'hidden' question here, if you would like to move your answer too:

stackoverflow.com/questions/1490728/... – **Benjol** Sep 29, 2009 at 5:37



Unloading projects is meant to be a temporary action so you can edit the actual project file as XML (text). If you

1



want to completely remove a project from your solution, you should use the "Remove" menu option, which will take care of removing any references to that project.



One advantage to using project references is that it allows you to easily debug through the code. It also automatically ensures that you are using the correct configuration build (ie, if you are building in "Debug" mode it will use the Debug version of the assembly). That being said, you lose some determinism about which version/build of the dependent project you will pick up - project references mean you always use the latest.

Yes, for Visual Studio to determine build dependencies it must be able to see and build all of the projects which would mean project references.

Share Improve this answer

answered Oct 21, 2008 at 20:30

Follow



[Scott Dorman](#)

42.5k ● 12 ● 81 ● 112



I've just had a eureka moment reading through MSDN doc on structuring solutions and projects.

1



What I hadn't noticed is that in a multi-project solution, the context menu in the Solution Explorer proposes a Project Dependencies popup. Here you can define the project dependencies manually, if you haven't defined them by project references between projects.

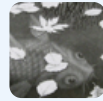


See [here](#) (MSDN link, so will self destruct after a few weeks)

Share Improve this answer

Follow

answered Oct 29, 2008 at 8:11



Benjol

66.4k ● 55 ● 192 ● 275
