

Do Visual Studio 2005 testing tools contain tools for mocking?

Asked 15 years, 9 months ago Modified 8 years, 10 months ago

Viewed 215 times



1



I'm working with VSTS 2005, and while the unit testing tools are fairly straightforward, I'm left wondering if there is any sort of support for mocking. I'd hate to have to do the mocking manually, because that leads to a lot of (mostly generated) boilerplate code.



I have an edict from The Powers That Be, that a third-party mocking library is not an option. I'm pretty sure that means I'm going to be mocking by hand, but one can always hope. :)

visual-studio

unit-testing

mocking

Share

Improve this question

Follow

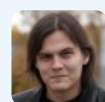
edited Feb 18, 2016 at 15:47



Daniel Mann

59k ● 13 ● 105 ● 127

asked Mar 10, 2009 at 12:12



Rytmis

32k ● 8 ● 61 ● 69

I tend to think I have it rough because we will go with inferior MS frameworks over better open source solutions, but then I hear things like this. I feel your pain man – [Matt Briggs](#) Mar 10, 2009 at 12:22

I'm running queries in the background to verify whether or not we have an officially sanctioned mocking library, but I'm sort of doubtful. The edict is against introducing third-party dependencies in general. At least we're doing unit testing. :)

– [Rytmis](#) Mar 10, 2009 at 12:34

Look, let's face it, The Powers That Be at your establishment are dumbasses for laying this requirement on you. The solution? Lie. Say MS is considering shipping Rhino or MoQ like it's shipping jQuery now. Keep track of time savings if your tissue of lies ever falls apart, tho! – user1228 Mar 10, 2009 at 13:18

No. TPTB are the customer. Trying to subvert their rules would be unethical, not to mention stupid. However, further probings have revealed rumors that a mocking library has been in trial use somewhere, and I'm currently furiously pursuing the possibility of getting the library in use for myself. :) – [Rytmis](#) Mar 10, 2009 at 18:28

It may help in understanding the situation if I add that I'm working on-site with the customer's (very tightly locked down, due to Good and Valid Reasons) equipment and software.

– [Rytmis](#) Mar 10, 2009 at 18:30

2 Answers

Sorted by:

Highest score (default)



No, there is no support. You would need to use 3rd party tools for this (oops guess you can't) I mean you will need to do it by hand...sorry to be the bearer of bad news

2

Share Improve this answer

answered Mar 10, 2009 at 12:14

Follow



Alex

13.2k ● 3 ● 33 ● 46



Mocking by hand it is, then. Damn. :(– [Rytmis](#) Mar 10, 2009 at 12:14



1



Having a requirement for no third-party libraries in production code may be a reasonable requirement (I would disagree), but forbidding you from using a third-party (and typically free) library for your unit tests seems insane. Developing your own mocking framework, unless you have requirements that aren't addressed adequately by the existing ones is a waste of valuable developer time. You ought to spend your time writing tests and code to accomplish your business purposes and a minimal amount of time writing auxiliary software to help you do that.

I know that there are people who feel that all mocking frameworks are evil and you should always hand mock. That's a different argument with some valid points, though I'm not in that camp. But if the issue is you can use a MS mocking framework (which doesn't exist, at least yet) or no framework at all, that smacks of stupidity. A mocking

framework used in unit tests won't compromise your production code.

Please feel free to share my response with **The Powers That Be™**

Share Improve this answer

answered Mar 10, 2009 at 12:39

Follow



[tvanfosson](#)

532k ● 102 ● 699 ● 798

The question is not of compromising production code, it's of consistency throughout the organization, which is necessary, for certain Good and Valid Reasons. The recommended approach is to write stubs by hand, but I'm going to make the case that unless the interfaces are small, it gets tedious fast.

– [Rytmis](#) Mar 10, 2009 at 18:32
