How to deal with an Undocumented API/Framework under .NET? [closed]

Asked 16 years, 2 months ago Modified 7 months ago

Viewed 866 times

3

Closed. This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet [Stack Overflow guidelines](). It is not currently accepting answers.

We don’t allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 6 years ago.

Improve this question

For work I have to code with an external company's API to deal with their proprietary database solution. Unfortunately the documentation they provide is more of an example guide then proper API docs, so it is very light on nitty gritty details like error codes, method returns, and exceptions.

So for instance, a class will have a `.GetErrorCode()` method, but I have no idea what those error numbers mean because they didn't document what number matches up with what error. In many cases a method will return an Object, with no documentation of what the type of Object it actually returns. I have asked them repeatedly for proper documentation but they seem to think details like the ones above are propriety secrets. So, is there any tools or methods I can work around my limited or in some cases non-existent documentation.

Please note that I am using Visual Studio 2005 and coding in C# under .Net.

And before anyone answers, "don't use the API", I have to, it is for work.

c# .net frameworks

Share

Improve this question

Follow

edited May 21 at 17:43

Brian Tompsett - 汤莱恩
5,875 ● 72 ● 61 ● 133

asked Oct 17, 2008 at 12:20

James McMahon
49.5k ● 69 ● 210 ● 288

5 Answers

Sorted by: Highest score (default)

6

A nasty scenario. I hate to suggest it, but maybe reflector is your friend if it isn't obfuscated. There may be some IP issues, but in this case reversing it seems the only viable way of finding out what the API is. However, I suspect (from methods like .GetErrorCode()) that this is a shim on top of P/Invoke - in which case even reflector will just show you lots of external calls...

The main other thing I can say is: write *lots* of unit tests that cover how you are trying to use it... that way if you guess wrong and something changes, you'll know early.

Share Improve this answer Follow

answered Oct 17, 2008 at 12:24

Marc Gravell
1.1m ● 273 ● 2.6k ● 3k

3

If I can't get code samples or talk to an original developer, I usually resort to Reflector to look at the underlying code. It's slow and inefficient, but sometimes that's all you can do.

Share Improve this answer Follow

answered Oct 17, 2008 at 12:24

Danimal
7,710 ● 8 ● 48 ● 57

3

It depends on your situation. If you're paying for the API, you should continue to pressure the company for better documentation of how to use the API.

If that won't work, what I would do is start my own documentation as I develop. Keep a notebook, a personal Wiki (screwturn wiki comes to mind), or some sort of electronic documentation. As some other people mentioned, you can use Reflector to help get to the source (if it's not obfuscated).

Creating your own documentation may not be what you were looking for, but if you can't get real documentation, at least create some as you learn things so that you can have something to guide you a few months (or years) down the road when you're trying to maintain the code built with the API.

Share Improve this answer Follow

answered Oct 17, 2008 at 12:33

Dan Herbert
103k ● 51 ● 192 ● 221

1

Best tools that come to mind include phone and e-mail. Hopefully you can keep bugging people over there until they give you more useful info... :(

Share Improve this answer Follow

answered Oct 17, 2008 at 12:24

Brian Knoblauch
21.3k ● 16 ● 61 ● 96

Use Reflector to see the source code. Should show you the Enum for the GetErrorCode message.

1

Reflector, by the way, is the greatest program in the history of the universe.

Share Improve this answer

Follow

answered Oct 17, 2008 at 12:24

Tom Kidd

12.9k ● 20 ● 91 ● 134