

How can I call a .NET form from an MFC application?

Asked 15 years, 8 months ago Modified 14 years, 1 month ago

Viewed 4k times



2



We have a several large MFC applications which presently call a COM object to bring up a complex dialog.

We would like to integrate the dialog into the applications -- we do not want to continue to use a COM object.

I'm investigating the possibility of building the dialog in .NET as a separate project (using Windows forms, not WPF) and providing a second C++/CLI project which calls it and which can be called from ordinary C++ code. This structure is so that the several applications that need to incorporate the dialog can just pick up the projects in their solutions. (The apps are legacy apps, and rewriting them extensively is not possible -- we're slowly moving them to .NET, but this is a multi-year project. Converting the apps to C++/CLI is not an option.)

I've built this and tested it from a model application, but so far I'm unable to get it to work in the simplest of the large apps, and based on some things I've read, I'm beginning to doubt that it is possible. (See [this link](#), especially. I'm aware of [this Stackoverflow question](#), but it does not seem to be relevant.)

So. Is this even possible? Any suggestions on how to proceed?

c#

.net

c++

mfc

Share

Improve this question

Follow

edited May 23, 2017 at 12:13



Community Bot

1 • 1

asked Apr 2, 2009 at 21:19



mlo

413 • 5 • 16

4 Answers

Sorted by:

Highest score (default)



2

Encapsulate your Windows Forms controls in a User Control and embed them in your MFC app using DDX_ManagedControl:



<http://msdn.microsoft.com/en-us/magazine/cc163605.aspx>



Share Improve this answer

Follow

answered Apr 3, 2009 at 1:07



drewh

10.2k • 7 • 35 • 43



Have you been successful in getting the C++/CLI project to call into your .NET component? I would try to narrow

1

down between which of the two layers it is failing to get through.



Since C++ code can call COM components, you could just compile the .NET dll with COM support. I haven't done COM in C++ though myself, so I can't tell you the nitty details. But I've made lots of COM exposed .NET DLLs and that side of it is fairly easy. Normally just a couple checkboxes in the Project Properties (i think under an advanced button in assembly tab).

Share Improve this answer

answered Apr 2, 2009 at 21:22

Follow



AaronLS

38.3k ● 23 ● 148 ● 214

We already have a C++-based COM component. The point of this project is to eliminate the need for COM. – [mlo](#) Apr 2, 2009 at 21:23

I get failures in the OLE initialization before execution ever gets near the code in question. Googling that error led me to the first link. – [mlo](#) Apr 2, 2009 at 21:26



Why do you need the middleman? Why not just

0

Native C++ app --> C++/CLI `class` `library`



The C++/CLI library provides a native wrapper around a CWinFormsView or CWinFormsDialog.



But, why do you need to eliminate COM? It's plenty fast and once you're good at implementing the interfaces the ceremony isn't too bad.

Share Improve this answer

answered Apr 3, 2009 at 1:05

Follow



Aidan Ryan

11.6k ● 13 ● 59 ● 87

There is a lot of C++ code in the existing COM object I'd rather not port to .NET. The structure I'm using has two complicated dialogs as separate C# projects called from the bulk of the code which is the old C++ code. That's called from the main app just as now. – [mlo](#) Apr 7, 2009 at 21:25

OK then I'm confused as to what the question is. Do you have this working or not? – [Aidan Ryan](#) Apr 8, 2009 at 2:24



0



I finally solved this problem with the help of a very good Microsoft support specialist. There were two problems, one was that the Boost library's threading is incompatible with C++/CLI in its default state. One solution is to compile with a different set of flags and then statically link it. The other is to use it as a dynamically linked DLL, which is what I wound up doing.

The second part of the solution is to set CLR threading in the linking property to STA, since OLE initializations fails with an unhelpful message if you don't.

Share Improve this answer

edited Oct 26, 2010 at 20:15

Follow

answered Oct 26, 2010 at 19:20



[mlo](#)

413 ● 5 ● 16
