

What's the best way to use .NET classes from Visual Basic 6?

Asked 16 years ago Modified 16 years ago Viewed 2k times



5



At my workplace I'm stuck with Visual Basic 6, but after reading the answer to "[How do I sort arrays using vbscript?](#)" I got my hopes up, that at least some of the missing VB6 features can be supplemented by .NET features. (Different kinds of collections, mostly)

It seems, that at least some .NET classes are available for scripting and can be used from VB6 through `CreateObject`. But I get a distinct to-good-to-be-true feeling, so...

- Are there any pitfalls I might encounter? (Besides that .NET has to be installed)
- Are all .NET classes available through `CreateObject` ?
- Is there any way to import those classes into my VB6 project, so I can have IntelliSense?
- Do you have any general suggestions regarding this approach?

.net

vb6

scripting

Share

Improve this question

Follow

edited May 23, 2017 at 12:04



Community Bot

1 • 1

asked Dec 18, 2008 at 7:43



Daniel Rikowski

72.4k • 62 • 256 • 336

You'll want to research "COM Callable Assemblies". There is a plethora of info on the INET about it. [Here's a start.](#) [This has some good info.](#) – ChrisN Dec 18, 2008 at 8:38

4 Answers

Sorted by:

Highest score (default)



6



There's lot's of information available at the [VB Fusion Developer Center on MSDN](#). Of particular interest will be [Using the .NET Framework Class Library from Visual Basic 6](#) and [Can I Interest You in 5000 Classes?](#).

You'll need to write some code in .NET to use the techniques described in the articles. If you don't already have Visual Studio 2008, you can download [Visual Basic 2008 Express](#) for free.

Share Improve this answer

Follow

answered Dec 18, 2008 at 14:52



Rob Windsor

6,859 • 2 • 23 • 26



From working on the conversion for my company CAD/CAM application. I do not recommend going from

2

the bottom up i.e replacing the core DLL or adding core functionality thru .NET.



I had much better results working from the top down starting by replacing forms and working my way down into the Core DLL.

The biggest hurdle was providing a suitable printer and graphics interface. But Microsoft release of a Printer Compatibility Kit solved both problems. Luckily when .NET was but a glimmer on the horizon I put all the VB6 centric graphic and printing calls behind a interface.

Again start by replacing the format with a .NET assembly calling your COMM core libraries.

If your logic is intertwined with the forms I strongly recommend separating it out into a new COM DLL IN VISUAL BASIC SIX. Have the new DLL interact with the Form through a interface. The form will be a thin shell passing events to the DLL and executing various operation (drawing, re-arranging, etc0.

If you can't avoid using .NET assembly to provide new functionality then create a series of well defined and limited interfaces to control the interaction between the .NET side and the VB6/COM side.

For example we wanted to add office/job management software for our industry to the range of software we sell. We decided to implement using the .NET framework. It had to interact with our CAD/CAM software and so we

create a series of interfaces where each can pull needed information or trigger action from the each other.

Finally if .NET classes are implement COM Interfaces the big gotcha is that you need to make sure that any parameters that are part of properties in COM must be declared ByVal or .NET will not be able to implement the property and fail to implement the interface. This includes the value parameter of a Let or Set

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

[edited Dec 18, 2008 at 13:22](#)

[Follow](#)

answered Dec 18, 2008 at 13:15



[RS Conley](#)

7,206 ● 2 ● 22 ● 37



1



Maybe [this](#) is helpfull:

By using the available information many of the classes found in the MS .NET mscorlib.dll have been reproduced in Visual Basic 6. This it NOT an implementation of the CLI runtime, only some of the unitlity classes that are in the mscorlib.dll library file.

Share Improve this answer

answered Dec 18, 2008 at 14:32

Follow



[dummy](#)

4,284 ● 3 ● 27 ● 36



0



You should only use interop as a mechanism for pushing data between systems - not for core logic code; the performance hit of all the marshalling etc will be too much. I suggest you look at using the more recent VB variants in VS2008 etc. It is a big shift, but if you can afford to make the change, you won't regret it.

Personally, when I moved from VB6 I switched to C#, but that decision has to be based on local context etc.

Share Improve this answer

answered Dec 18, 2008 at 8:34

Follow



[Marc Gravell](#)

1.1m ● 273 ● 2.6k ● 3k

Down-vote all you want, but using .NET collections etc as routine from VB6 *really* isn't a good idea. – [Marc Gravell](#) Dec

18, 2008 at 14:27
