

The Only Add-In Always Used

ONLINE ONLY

There's only one add-in that I have auto-loaded every time I fire up the Classic VB IDE. After disappearing from the market, it's back now, and it's free!

June 15, 2009 - by Karl E. Peterson

I can't count how many times over the years I've seen folks express "really weird" issues in the Classic VB IDE, and how often these can be solved with the simple advice: "First, unload all add-ins, then restart the IDE." As much as these little tools help, they can almost as often confound. This is unfortunate, because some of them are real jewels.

I'm sure everyone has a favorite add-in that they just can't imagine living without. ([Leave a comment](#) at the end of this column if you'd like to share your favorite.) My own is [vbAdvance](#), written by Peter Young. I have it configured to load every time I start VB5 or VB6. No other add-in in my arsenal is similarly enabled, and honestly none other is so critical to the product I produce.

Most add-ins are oriented toward specific IDE tasks -- formatting or analyzing code, constructing forms or positioning controls, and so on. In contrast, [vbAdvance](#) is all about the resulting binary file. It injects itself between the compiler and linker to modify the final EXE or DLL output, thus opening up options otherwise not available to ClassicVB authors.

For example, I write a lot of console applications. Without [vbAdvance](#), you have to resort to post-processing the executable to [toggle a bit in its PE header](#) or [relink after compilation](#). Both are tedious steps, in comparison to simply marking a checkbox in the [vbAdvance Settings](#) dialog. From that point onward, this project will always be linked such that Windows recognizes it as a console application rather than a windowed one.

I've also used [vbAdvance](#) to devise custom `DllRegisterServer` and `DllUnregisterServer` functions, so that when my DLLs were registered or unregistered, I could control what other activities simultaneously took place. This allowed me to implement a trial period check algorithm, whereby the time of first registration was stored away in a "very secret" place so that it could later be checked against the agreed on trial period. The failure to provide for exported functions for tasks such as this is one of the biggest omissions in ClassicVB, but [vbAdvance](#) fills this need nicely.

There are numerous other settings "that VB forgot" you'll come to love. Specifying an application icon for a formless application is just another dialog entry. Need startup or teardown notification in your DLLs? It's there via an exposed `DllMain` entry point.

Another checkbox will automatically inject an XP manifest into your EXE, and you just need to add a single call to `InitCommonControls` to insure support for XP themes. Want to write a Control Panel application?

Unfortunately for Pete, I'm what might be termed a niche customer. Apparently, mainstream Classic VB folk concentrate on GUI database stuff. Go figure. Bottom line is, vbAdvance was simply never the revenue-generation machine it needed to be to sustain the sort of support burden offering these esoteric functions turned out to be. So, after several years as a commercial product, the white flag was waved and it became an unsupported freeware offering. Time passed, the domain expired and vbAdvance pretty much disappeared from the scene.

This development disappointed me, as people continued to ask how to do things I knew were incredibly easy with this powerful tool. One day, a couple of months ago, I wrote Pete an e-mail asking how things were, "whatever happened" and so on, and, well, long-story short, he has agreed to allow me to host downloads of this marvelous add-in on [my Web site](#).

It's 100 percent freeware, and 100 percent unsupported.

But there are folks who have a fair bit of history with it and would likely have ideas if you run into problems. Just swing by the [microsoft.public.vb.general.discussion](#) newsgroup to find the diehards. There are also a few handy demo programs that come with the download, including a complete [drop-in ready Console module](#) I wrote.

If you've been reading my columns for years, I'm pretty sure that you, too, fit the profile for vbAdvance -- [go download it!](#) Even if you don't find immediate utility in it, the day will come when you can't do without one of the many enhancements it brings to Classic VB. It definitely belongs in your toolbox.

About the Author

Karl E. Peterson wrote Q&A, Programming Techniques, and various other columns for VBPJ and VSM from 1995 onward, until Classic VB columns were dropped entirely in favor of other languages. Similarly, Karl was a Microsoft BASIC MVP from 1994 through 2005, until such community contributions were no longer deemed valuable. He is the author of VisualStudioMagazine.com's new [Classic VB Corner column](#). You can contact him through his [Web site](#) if you'd like to suggest future topics for this column.

1105 Redmond Media Group

Copyright 1996-2009 1105 Media, Inc. See our [Privacy Policy](#).