PRODUCT REVIEWS

VISUAL EXPERT

"In the Real World"

Bill Aumen

About the author

Bill Aumen is an independent contractor, providing turn-key solutions, and an enthusiastic PowerBuilder developer. He can be reached at bill.aumen@island.net

E-NOV@ CLUB

Introduction

I’m a believer… a Visual Expert believer that is!

I develop and support a number of PowerBuilder applications, with the assistance of contract developers. Each developer is a world class PB programmer, but the fact that we are not constantly involved in any single application means we need a “refresher” each time we start a maintenance project.

Recently we had to make some major enhancements to a PB system. This is the type of task where the carefully planned inheritance structure of the system usually goes to pot. You see some code that is almost what you need, but you duplicate rather than extending it, for fear of fouling up something that is already running well.

I had used VE for a year and a half already for small jobs, but this seemed like the perfect opportunity for me to really put the tool to the test.

My Favorite Features

In my book, the top 2 VE features are the ability to: 1) easily view and navigate the object hierarchy, and 2) easily see the code that will be affected by our changes.

First I browsed the application with VE to get a clear picture of the object structure. I formed a list of the objects that needed to be created, and those that we would extend for additional functionality.

With that list of modifications in hand, I used the Search capability to review the "where used" list. What else would our changes affect?

That search produced my first surprise: we had very carefully designed the object structure when we created the system. But right here before my eyes in the search view I could see redundant code!

Redundant code may not surprise many of you, but since I continue to support each of the applications we develop, I pay the price for sloppy work. And I hate to make the same change twice when there are so many more fun and creative things I could be doing! We were definitely going to clean this up.

Favorite Features, continued:

Numbers 3, 4 and 5 on my VE “favorite features” list are: documentation, dead code candidates, and duplicate objects.

Documentation Centre
We employ peer “code reviews” along the way to balance the priorities of the project with the quality of our code. Looming deadlines tend to produce sloppy code.

For this purpose I still prefer printed documentation that I can browse through and markup. There are so many options in the documentation that it took a little trial-and-error to produce exactly what I needed the first time. Since VE saves those settings for me, it is now a no-brainer.

The generated documentation makes a wonderful, and useful, package to accompany our software. I will be the first one to admit I never take the time to update static documentation with each program change.

Dead Code
I find the ability to quickly see “Dead Code” candidates very reassuring. We strive to maintain clean code, but each cycle of enhancements increases the likelihood of no-longer-used objects and code. Novalys goes out of their way to point out they only detect static references. This is a good point to keep in mind, but from a practical standpoint I always consider these Dead Code “Candidates”. Easily viewing the code, coupled with the Search feature, lets me identify the objects that can be deleted.

Duplicate Objects
How nice of PB to permit this feature. How confusing it is to debug when I did not *intend* to create a duplicate object!

Once again, VE makes it effortless to see the duplicate objects.

Version 5.5
I liked VE 5.0 just fine. And 5.5 is definitely nicer to use.

You can see from the Change History on Novalys’ web site that version 5.5 contains incremental improvements, not revolutionary new features.
- The shortcut keys are now much more intuitive (Ctrl-F for find, instead of Ctrl-R, and others).
- Search now automatically saves a View for further reference. (Guess I was not the only one who closed the search results before getting all the info I needed ☺.)
- A “history” feature, similar to Internet Explorer has been added.
- And more…

Should you purchase the upgrade?
- There are no major new capabilities over 5.0
- But there are some very nice improvements
- And, most important for me, 5.5 supports PB 10.

What Would I Improve?

First, there is so much information in the GUI that it can be overwhelming. I would like to see cleaner screens, with further details available in popups or context sensitive menus (Having said that, I had the same problem with the new interface beginning with PB 7. Now it is second nature.)

Second, it would be nice to see the generated documentation easier to print. The Minimum System Requirements for VE include Microsoft Word, so the use of Word's "master document" feature could reduce the effort of printing the individual documents that are generated. It is a minor point to be sure.

Conclusion

At the end of our project, we had *improved* the object structure. Our user Beta test cycle was a breeze: because we now had better code than when we began the enhancements, it was far easier to locate and correct bugs.

Delivering enhancements that don't create more problems than they solve makes us look like heroes.

Thanks Novalys for packaging so many useful features in this tool. The end results for my projects are: cleaner code, easier maintenance, and nice documentation.