
Subject: Re: The Book

Posted by [davidf](#) on Thu, 22 Jul 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Ritscher (david.ritscher@bigfoot.com) writes:

- > I have a feeling I might have been one of the 57 people you mentioned
- > who ordered your "IDL Programming Techniques" book on the same day.

I don't want to give you the impression this happens every day. Lord knows if it did I'd be playing more tennis in a futile attempt to improve enough to keep up with my hot-shot son. (I beat him 6-1 tonight, but he said he was "tired". Uh, huh. Getting my hopes up, more likely! :-)

- > I have received it, and have been really enjoying it, as well as the
- > associated examples. It is a nice piece of craftsmanship, both
- > aesthetically and technically.

Thank you for this. Sometimes Dick and I wonder if the craftsmanship ever gets noticed, but we go to a fair amount of trouble to try to get it right. I appreciate your comment very much.

- > Could I put in a request for your next book? I
- > would entitle it: "The Coyote Cookbook: an IDL Project
- > Building Block System." It would come with a complete
- > set of building blocks,[...] which one could easily build
- > up an application.

Uh, oh. The cat is out of the bag now. You realize, of course, that this is **exactly** what I had in mind for this book from the very beginning, but thought it best not to advertise it that way too heavily. My courses are even more that way, although I have to tailor them sometimes more than I would like to stay within the prescribed RSI curriculum.

But, yes, these example programs (in fact most of the programs found on my web page) were designed to be the building blocks of applications. Dick and I use them over and over again in our consulting work. (How else can we put together an elegant prototype application in just two days?) The book is really about some simple ideas that when used together can be quite powerful.

I like your suggestion (although finding time to implement it seems impossible to me now), but you would only have

to follow me around for a day or two to know how much extra work it would entail. Although the programs are good, they are never *quite* right and Dick and I spend many hours tweaking them to get them to work the way we want them to in client applications. Sometimes all we use are the *ideas* behind the programs and often we just steal the event handler code. We would never hear the end of people pestering us to do this or do that with the programs. And then, my history of getting programs to work right after I fix them is... well, let's just say I spend way too much time ftping things around. :-(

I was hoping we could get some kind of collaborative group thing going here for a library of routines, although I think it unlikely unless someone steps forward who is willing to badger and prod the rest of us for a chance at personal glory. (I was sent, by the way, some *very* thought provoking code by a person who knows IDL well that I want to share with the group. I'm only waiting for a bit of time to organize it before I make it available.)

- > In a way, it would be an alternative
- > to IDL's GUI builder - or maybe I should say the opposite of, since it
- > would consist of content, instead of an attempt at the outward form.

Yes. A friend of mine and I spent quite a bit of time about a year ago building a GUI application builder. It was like RSI's, although it did have code recovery built into it, so it could be modified over and over again. But as part of that experience we thought a lot about a program "template" and what that should look like and how the pieces of an application should work together. I think we both came to the conclusion that what we wanted were small, very specific tools that could work effectively together.

- > Example building blocks would be a file selection menu, a plotting
- > tool like XWindow, a floating point slider, object data handling
- > tools, and numerous other small blocks, that could be assembled easily
- > into an application where the data flow is handled well and the whole
- > things integrates into a clean application. Thus it would get people
- > up and running faster at building mid- to large-size projects.

What I realize now is that if I were going to build such a library, I would be building it with objects and direct graphics calls. Objects because they are clean, elegant, and can be quite "smart". Direct graphics because I want them to display and print fast.

I appreciate these ideas, and I don't want to disappoint you, but I wouldn't hold your breath waiting for the book. Better

to take what is out there now and make the best of it. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155
