

March, 1996

Editor's View

Choices and Changes

Giving you the best magazine involves some hard choices.

By Tamar E. Granor, Editor

Code or text. Techniques or tools. When we put FoxPro Advisor together each month, both of these areas need to find an appropriate balance.

In the last year or so, finding that balance has become much harder. Far more tools worthy of publication have come to our attention. Many articles, both tools and techniques, seem to have more code than ever before.

Until now, unless a tool was in the public domain (like GENSCRNX), we've always tried to print all the code needed to make the tool run. In the best situations, the code itself contained techniques worthy of discussion. But that's not always the case, and we're now finding this is a limit we can't live with. It eliminates too many useful tools, and devotes too many pages to code that doesn't teach you anything.

Beginning with this issue, we include some articles about useful tools where the tool itself is provided only on the Companion Resource Disk. In the space we save by not printing the code, we'll be able to include more useful techniques (with enough code to get them running).

If you don't subscribe to the Companion Resource Disk, you can order individual disks when you want to put one of these tools to work for you. See the staff page for ordering information.

Even more than in older versions of FoxPro, tools are an essential part of working in Visual FoxPro. We believe this new approach allows us to serve the FoxPro community better than before.

OOP or Oops

Lots of people are trying to figure out how Object-Oriented Programming fits into their development cycle. Most people get the idea of properties and methods. But, lots of people are having a hard time getting beyond that point.

In fact, there was quite a lively discussion in CompuServe's FoxUser forum a while back, asking whether insistence on object-orientation is making the learning curve for VFP too steep.

One of the traditional strengths of most XBase languages has been that you could try things in the Command Window (or its predecessor, the dot prompt) and see how they work. Once you understand, you can move the concepts into a program.

Apparently, the focus on VFP's OOP aspects is keeping some people from taking the same approach to learning VFP. It shouldn't. In fact, it's easier than ever to fiddle from the Command Window. Unlike older versions, in VFP, once you display a form, you can get to the Command Window and issue commands that change the form you're looking at. You can even start with an empty form, add controls to it, tweak the properties of the form and the controls, and then use the form's SaveAs method to store the form you've concocted for later—all from the Command Window. (The secret to messing with an existing form is the `_SCREEN.ActiveForm` object reference. Use it to get to the form's properties, events, methods and controls.)

In the forum discussion, it came out that what most learners find hardest is the notion of inheritance. I see evidence of this regularly when people use the Form Wizard to create forms and then ask "where's the code?" The idea that you call a routine (a method) in the program you're running and it's not actually located there, but in some other program, yet it gets run properly, is pretty wild.

But inheritance is the key to being productive with OOP. Here's a short, but true, story I heard third hand. A developer from one of the better known companies wrote an application for a client in Visual FoxPro. The initial spec stated that all text boxes should have the "select on entry" feature. When the app was nearly done, the client decided they didn't like "select on entry" after all and wanted it removed from every form. Big job, right? Nope, the developer had subclassed the base textbox class, giving it a Format of "K" (select on entry). All textboxes in all forms were based on that subclass or one further down the inheritance chain. All the developer had to do was go back to that one class and modify the Format property. Inheritance saved hours of tedious work.

How do we help everyone climb the learning curve? Obviously, we'll continue to publish articles at each level. (See Miriam Liskin's columns in this issue and the last one and Drew Speedie's article this month for some perspective on the OOP issue.) But perhaps all of us who've moved farther up the curve need to be more sensitive to this issue and make sure we don't project an attitude that says, "There is only one true way." (In fact, that's probably good advice in our everyday lives, as well.)

More Changes

You may have noticed two new Contributing Editors listed the last few months. We've added regular columnist Miriam Liskin and frequent contributor Nancy Jacobsen to the roster of experts who give us guidance. Miriam's ability to explain complex concepts in simple terms is practically legendary. Making sense of interfaces is one of Nancy's many skills. We're very pleased to have both of them on our team.

In addition, beginning with this issue, Patrice Hartmann takes over as Managing Editor. AnnMarie Garcia, who's done a terrific job over the last year and a half, is moving on to other challenges within Advisor Publications. Patrice has worked on Clipper Advisor and VO Advisor, as well DataBased Advisor, in her time at Advisor. Before that, she worked at New Riders Publishing. I know she'll continue to hold FoxPro Advisor to the high standards established by her predecessors.