September, 1998

Legacy Apps and Year 2000

What do you do when clients have applications that do just what they want, except that they're not Y2K compliant?

By Tamar E. Granor, Editor

In the March '98 *FoxPro Advisor*, we ran an article about a very clever solution to the Year 2000 problem for FoxPro 2.x applications. Christof Lange's code takes advantage of a command few of us ever noticed to set up a rollover feature like the one in VFP 5 and later and, thereby, allow users to continue to enter two digits for the year. Clearly, this is a hot topic in the FoxPro community – we've already sold out the Professional Resource Disk for that issue. (If you still need a copy of Christof's code, you'll find an updated version on this month's PRD.)

Christof's solution is best suited to applications built using the FoxPro 2.x Power Tools, since it can be automated in those situations. It also requires that the application's source code be available.

I have a client, though, who has a number of FoxPro 2.0 and FoxBase+ applications, none of them built with the Power Tools. All of these applications were custom designed and written for this company, do exactly what they need, and, in at least a couple of cases, there's still nothing available off-the-shelf to do the job, though they've been using these applications for close to 10 years. All of these applications have Y2K issues.

Though it would be a lot of work, I could apply Christof's solution to the FoxPro 2 applications. I guess I could even move the FoxBase+ applications into FoxPro 2. But I didn't write these applications, and, in this case, I tend to subscribe to the "if it ain't broke, don't fix it" philosophy.

Through one of those wonderful coincidences, literally days before the client and I realized there was a problem, I received a message from Neil Weicher of Communication Horizons (the folks who make NetLib). He'd been following various Y2K discussions in CompuServe's FoxUser forum and had come up with an innovative solution.

Y2KFix is a TSR that you load prior to running your application and unload when you're done. Like Christof's solution, it lets you specify a rollover date – any years entered that are less than the rollover are transformed to 21st century dates, while years greater than the rollover stay in the 20th century. Unlike Christof's solution, Y2KFix requires no code changes in most cases.

Y2KFix is available in both DOS and Windows versions. Neil is even working on FoxBase+ support in the DOS version, though that will require using LOAD and CALL and, therefore, may need source code. (I say "may" because someone has suggested that it might be possible to do the LOAD and CALL in a new program that then calls the original EXE.) For those of us supporting legacy apps that still do what the client needs,

the ability to keep them in FoxBase+ is a real bonus. (I have vivid memories of moving a client's FoxBase+ application into FoxPro 1.0 not long after I got into this business, and having to deal with a lot of visual issues.) For more information about Y2KFix, contact Communication Horizons at www.netlib.com.

Y2KFix is not the only product available to help the Fox community solve Year 2000 problems. Several companies have products that help you find your Y2K problems and get you started on solving them. FoxPro Advisor has carried ads for Scanalyzer 2000 from Hendela System Consultants (www.scany2k.com) for some time now. I also heard recently from Clyde Getty of Getty Information Systems about his FoxPro Project Inspector, which appears to perform a similar task.

I haven't tried either of the scanning products yet. My hope is that Y2KFix will handle my clients' fragile applications. If not, checking out the other products will become a high priority item for me.

Beyond these old applications, I'm being very careful to see that anything new I write doesn't have Y2K problems. VFP 6 makes this easier than ever with its new SET STRICTDATE command that slaps me on the wrist for any use of ambiguous dates. Look for an article in a future issue to show you how to make sure your VFP applications don't run into Year 2000 problems.

My New Notebook

In the April issue, I mentioned that I was about ready to buy a new notebook computer. Sure enough, those of you who came to DevCon got to see my brand-new Toshiba Tecra 550CDT. Following my usual plan, I bought a machine near, but not at, the top of what was available. I find that approach generally saves me a lot of money without costing me much in capabilities.

My new Tecra is a Pentium/266 with a 4 GB hard drive. I loaded it up with 96MB of memory–the machine comes with 32, but the additional 64 MB wasn't terribly expensive. It's capable of 1024x768 video and boy, is it pretty. It also has the excellent keyboard I've come to expect from Toshiba notebooks and the eraser-head type pointer I prefer. Best of all, the bottom line was in the $4000 range rather than the $5000 I'm used to spending for a notebook. It looks like notebooks are following the lead of desktop machines and becoming less expensive. That's certainly good news.