

From the Editor

A New Year, Old Apps, and My Toenails

Whil Hentzen

The start of a new year is often a time to reflect on the year just ended. I had a jump on this activity because my January column was due in late fall, but nonetheless, I've been thinking about "things." One of those "things" has been my toenails. [Eeeeew! How icky!]

One Saturday morning in June about 15 years ago, a bunch of friends and I did a 20-mile run. I had the misfortune of wearing a new pair of shoes that hadn't been broken in well enough. I lost the toenail on my big left toe a week later due to a blood blister underneath the nail, which destroyed the tissue holding the nail to the toe. Later that summer, I delivered my largest-to-date dBASE III+ application.

At the time, all of my toenails and all of my applications basically looked the same. Sure, they were different sizes, and their shapes varied somewhat, but you could tell that they were all, er, developed by the same person.

Fast forward a decade and a half. Thirty thousand miles later, I'm in the midst of recovering from the worst injury of my life. It's been a long haul, and I'm not out of the woods by a long shot, but I can usually run four miles without the customary shooting spikes of pain through my shins. I've been examining my toenails as part of the ritual of putting my shoes on, and it's dawned on me that they're all different now.

On my left foot, the nail on the big toe looks like Rocky Balboa's face at the end of the first *Rocky* movie. The next nail is curved over the toe like the hood of a sports car. The third nail falls off every few months due to, well, I'll spare you the details of exactly why. [I'm getting a little queasy, Whil... and here I thought I was sparing you from the details of the really bad toes... <g>] The fourth is twisted around as a result of being squished by the toe to its right, and the fifth—well, I think I've been graphic enough already.

I'm not complaining, mind you, and they usually don't hurt. It's just that over the past 15 years, they've all taken their own special turns due to one event or another, and they're all looking a little worse for the wear.

My applications look the same way. They started out similarly but with the passage of time, despite my trying to keep them all in sync, they've diverged.

There's the PC inventory application that's been running since early 1992—yes, in FoxPro 1.0—with nary a peep, until a month ago when the server went down, taking all of the data files with it. We didn't even have the libraries that this app used loaded on any of the machines in the office. A week later, another customer decided to move a document tracking system to a different server, but missed one of the support files. Which support file? This app—FoxPro 2.0 vintage—had also been backed up and taken off of our network, but fortunately the structure was similar enough to the 2.6 stuff we're still working on that the diagnosis was relatively easy.

Another application is now on version 5.0 and it's showing its age. It was originally written in 2.0 and then got upgraded to 2.5 (still in DOS), then 2.6 in Windows. While the application is still working well, we've changed our attitude about certain design aspects, and yet have to deal with converting this older architecture.

So what's the point? I've been wondering what I'm going to do about these toenails that have all gone their separate ways—and these applications that have done the same thing. Do I invest the time into bringing them back into line so they all have the same look and feel, the same mechanisms, and so they all look like they're from the same place again? Or do I just let them stay as they are, not bothering to change them until there's a significant value in doing so?

Interesting how two seemingly completely different situations have a lot more in common than you would think, eh? In this case, the answer of what to do with my toenails has come from our customers. In many of these cases, the occasional problem can still be fixed by a smaller investment than it would take to rewrite or convert the application to our newest architecture. Same thing for my toes: They don't give me much trouble, and the occasional Band-Aid is a significantly smaller investment than the alternative.

Instead, I've found it more worthwhile to come up with a strategy to deal with the odd happenstance that causes a problem. With my toes, I spend a little bit of maintenance time each month and that reduces the likelihood of a bigger problem.

With the applications that have been out there awhile and have an old architecture in place, we're doing something different. Each time I (or one of the other developers in my shop) have to get into an application that we haven't looked at for a couple of years, we've found we have a few of those "What the heck is this?" type of questions.

When we were doing all of our applications that way, the basic structures and mechanisms were obvious, and thus, undocumented. We didn't need a set of instructions to tell us that the user file was named ZUSER.DBF and that the data dictionary files all began with "SYS" and were located in the METADATA directory. Four years (or even nine months) later, this information isn't as easily remembered, nor it is quite as obvious as it seemed it would be.

The answer to this problem was to create a log file for each legacy application. In it, we document each of those ridiculous questions that we'd never thought we'd have to ask, and the answer to the question as well. Sort of like one of those "FAQs" (Frequently Asked Questions) that you see here and there.

That's our strategy and we're sticking to it until a better one comes along—or until we stop changing the way we do things.