

My Vision



Carl Franklin
writes the Q&A column for
VBPI and is co-owner of
Carl & Gary's Visual Basic
Home Page (URL: [http://
www.apexsc.com/vb](http://www.apexsc.com/vb)).
Reach Carl at
CarlF@apexsc.com.

**IN THE EMERGING
INTERNET MARKET, VB
DEVELOPERS HAVE THE
ADVANTAGE—BUT YOU
BETTER BE READY FOR
IT. IN OTHER WORDS,**

On Snooze
GoTo Lose

Opinion

Who drinks coffee at 3 A.M.? Me, that's who. You got a problem with that? I've been a little edgy lately. I've been staying up late trying to achieve expert status regarding Internet application development, and in the process I'm single-handedly supporting a small village of farmers somewhere in Columbia. It isn't easy to digest all this hoo-ha, but believe me, this is the big one!

Let me explain. You're impressed by the size of the Visual Basic add-on tools market? The Internet development market will be 10 times hotter. For once, the software industry has a huge, growing, unified, clearly defined market: namely every Tom, Dick, and Achmed with a PC and a modem. Every one of 'em can look at your software. You didn't even ask them to. Distribution? What's that? I'll tell you what it is, "Press here to try and over here to buy!"

I'm talking about how software will be written, marketed, sold, and used in the not-too-distant future. Some of it has already started, but the real tidal wave has not yet begun. Client/server development as we know it is going to change. Back-end servers will change and we'll see more client platforms, but I believe the primary development model will be the Web browser and Web server. Even if the Internet isn't used for communication all the time, the model works on any scale.

Allow me to compare how we bring software to market today versus what's coming. For the sake of argument, let's pick corporate software—systems that will be used by major enterprises here and abroad.

To create an application, the state-of-the-art approach is to use VB, or (let's be fair) some other client/server application development system. On the back end you'll probably have a database server such as one or more NT-SQL boxes. The process of buying, writing, and using components has kept this magazine thriving because of the diverse options available to developers. But it can be time consuming. Don't expect any less confusion in the new market.

How will you distribute your system? Native database driver? ODBC? TCP/IP? CD-ROM? Floppy disks? Duplication machines? Manuals? Shrink-wrap? Distribution? In-house? Out-house? Replication? SQL update scripts? Version control? Tech support? Bulletin board? How many phone lines? How many machines?

Aaaaaaaaaaugh!

I make it sound difficult because it is. Sure, you'll be rolling in dough once you start cranking out the software. So you can afford such wasted resources. But all this will become futile soon enough.

Let's take a peek at the next frontier in computer software. Say we're going to write a hospital billing system for an HMO (just for you, Andreas). This app needs access to the complete billing history of each of the hospital's patients.

The HMO has branches all over the United States. Joe Patient used to go to the Cleveland branch, but he moved West, where he now uses the Yuma branch. When Joe gives his name to the teller in Yuma, the teller is going to use your application to pull up Joe's billing history from his visits to the Cleveland branch.

The teller asks him which other branches have treated him, and Joe replies, "Cleveland." The teller dials the Cleveland branch with a modem, uses your application to download Joe's file, and merges it with the local database. You might want to use a central repository for the HMO's patients, but then accessing it with a modem would take too long. Management of the servers and data at all these hospitals is a demanding task that can sometimes seem impossible.

Enter the Internet. Now you write a "download-and-run" Internet app that can be accessed with any Web browser that supports VB Script. Each hospital runs its own Web server. The user looks up a patient at the central server, which immediately links to that patient's hospital to let you access his or her data.

Manuals? HTML pages. Important tips and news? HTML pages. Distribution issues? None. Tech support? A link to an e-mail address, or even an embedded link directly to a tech support person. Updates? Couldn't be easier. Copy a few files, set a few flags, zip-bam-boom. Done!—for the entire world. Now that's awesome.

Even more incredibly, now you can introduce into your software something that's never been done before: dynamic advertising. How much do you think you could charge a pharmaceutical company for one month of advertising that would be seen daily by the 10,000 hospital employees in your entire enterprise? That's where we're headed. And if you've been programming in VB for any length of time, you'll have plenty of golden opportunities in this brave new world. I can't wait. That's why it's now 4 A.M. and I'm still awake (gulp, gulp, gulp). Shh—don't tell my doctor. ☒