
Moving to Windows

© 1995 by *Peter Neuendorffer*

DoTheNewMachine

It all began when my friends told me I had to write Windows software. I was sure the windows boom was a conspiracy to keep DOS writers down and out. Besides, where would I get the money for a 486 machine? When I applied for a grant, it was turned down. They told me that my work had to appeal to a certain group of users, to fit a specified criteria - and it didn't. So I pestered my local bank, and after some credit shuffling, I got a loan.

Almost immediately my three computer-literati friends started giving me advice: Do not buy *the* machine from a store. So I bought a machine from VTECH. They said I had to have 8 megs, and large hard drive, a .28 monitor -the works! So I sacrificed the multimedia and the printer because at last I would be making bundles of money- writing Windows programs.

When the new machine arrived and I unpacked it , I sat there using the box as a footstool, stupified, as I did when I got my brand new 286 several years before. The next couple of weeks were a haze of buying, selling and trading components. The Intel 14.4 modem that I bought, a great modem I'm sure, refused to work with ProComm Plus, so I got a Zoom. My ancient printer would not work under Windows, until the local Radio Shack guy let me know I needed a new cord. In fact, when I finally reached Tech Support , they told me the old cord might fry my mother board. I got Visual Basic and was off to the races.

The first thing I discovered is that it was fun to click windows on and off, push them around poking around a system that seemed bottomless. My very own DOS applications looked really dismal by comparison with their black screens. DOS became an orphan until I discovered that I didn't like File Manager at all. Give me the old DOS directory and file commands any day. After playing Solitaire, dumping the Windows tutorial, and adding some notes to Notepad, I started in with Visual Basic. I should also say that I had obtained an Internet account through Channel 1 in Cambridge, MA., expecting to go surfing any day.

DoTheOldLanguageTheNewLanguage

For me, working in Turbo Pascal had been like working for an old friend. This DOS based language was dependable giving me control of just about anything. If the language didn't do it, I could make a request of DOS with a pseudo interrupt.

I could put a character wherever I wanted it on the screen, and copy or paste portions of the screen with the greatest of ease. Instead, I had to do everything myself. Lists had to be sorted (Bubble, bubble), menus created (workhorse), a string input handler created (this thing I used like sourbread, and changed it for each new application). Putting nice and colorful text on the screen was a general pain:

1. Type in arcane incantations like `gotoxy(x,y);`
2. compile the program;
3. run the program.... oops wrong! Do 1,2,3 until drop dead from exhaustion.

If I wanted to fix a file, format a file, do anything with a file, Turbo Pascal was great. Some of the things I did: boolean text parser- search for and, or, not; numeric expression parser $x^{2/3}+b$; directions finder for Boston Transit; personal information manager; idea program; search for text on system. This last I turned into a chat program where you chat with your hard drive. Certainly screwy!

Do Visual Basic

Okay, so I'm a Windows beginner. Have pity on my incorrect information and approach. I have already seen a lot of Windows beginners programming on the Internet in the `comp.lang.basic.visual.misc` conference. These people, unlike me, the "expert", have never written a line of computer code in their life. They have the laudable "telephone" or "vcr" approach. I push this button, and it should do that. That is all very well if the state of the art of computer programming was up to this. Computer programming is not a trip to the ATM machine, although it should be.

Visual Basic is event driven. The analogy is that in the old procedural language, first you do this, then that, and then maybe the other thing. It's like a list of tasks that is sometimes logically complex and twisted. But the event driven language uses pretty graphical objects like a List Box or a Text Box (for user data entry.) When the user clicks on the object, the object's code is triggered by this event. That means that all I have to do is fill in the blanks, and I have a program.

Hold it. Not so fast. All the actions of the program have to coordinate. You get what I call the oops! effect in Visual Basic very fast if two events conflict, or a message doesn't make it from one area or "scope" to another. But this type of setup is great for fast development, and features like automatic sorting of List Boxes, and drawing your screens as you see them, without compiling the program, are wonderful tools.

So far I have written a dictionary program for Windows. It doesn't seem like it does all that much, but actually it does a lot very well. Now if I could just get the thing to make products without me having to sit there and click the mouse!

DoComments

The old DOS computer language: -

The good; Challenging and learn real programming the hard way.

The bad: slow, cumbersome, aggravating, keyboard tendonitis

The new Windows computer language: -

The good: Fast, click the mouse, Windows services, see it now

The bad: Fast, get into trouble very fast, Create big horrible programs fast, mouse complacency, let's not work too hard now.

Yesterday, I managed to blow away a computer reviewer's Internet mailbox when I sent him a file attached to a message. Now why didn't someone tell me? And then one of my DOS programs doesn't do too well when installed with a Windows icon. Why didn't someone tell me? But, all in all, Windows is better.

I can highly recommend two books on Visual Basic, which I am dutifully studying. Also the Internet conference `comp.lang.basic.visual.misc`.

Visual Basic 3 by Example, by D.F. Scott
Prentice Hall/Que, 1994, ISBN: 1-56529-724-5

Visual Basic SuperBible Second Edition, by Bill Potter
Waite Groupe Press, 1993, ISBN: 1-878739-50-6

Although a new Windows programmer, Peter Neuendorffer is an experienced DOS programmer. . His best known work finds routes on the Boston Transit System. He is a regular WindoWatch contributor.