

VISUAL BASIC MENU POTATO

VISUAL BASIC MENU POTATO

VISUAL BASIC MENU POTATO

(Is there an echo in here?)

Program, code and documentation ©1994 by David Stewart
Rev. version 1.1, released 20 December 1994

Have you ever realized that you needed a menu on one form that already exists on another form? If so, then you know the hassle you face in reproducing the menu: Either you create it anew by hand in VB's Menu Design Window, or you fuss around with Notepad or other text editors, copying and pasting menu code from one form text file to another.

Well, the hassle is over. Now you can let VB Menu Potato do the heavy lifting, while you sit around and click the mouse a few times.

VB Menu Potato is a simple program that makes it easy to reproduce Visual Basic form menus without having to recreate them manually in the Menu Design Window (arguably the least flexible and navigable of VB's IDE windows). It is part of the VB Potato series, a collection of VB programming tools available for very low prices as shareware.

VBMP does not require that you be working in Visual Basic when you use it. In fact, it is best if you are *not* working in VB, since you don't want to copy a menu from one form into a form you're already working on in your open VB project. (Fear not, VBMP won't let you. More about this later.)

Installation

VB Menu Potato comes complete with a manual installation program: You copy the files where they belong. The main program file, MENUSPUD.EXE, can go anywhere, but I think the VB directory is the best place for it to be. The custom control files, THREEED.VBX and SSFORMFX.VBX (Sheridan's Designer Widgets FormFX control, which accounts for the cool window novelties--note that you can drag the window by clicking and holding on its client areas as well as the wacked-out title bar; also, the form looks resizable, but isn't), you can copy to your Windows SYSTEM directory--but *don't* put them there if you have more recent versions of these files! If you have, just ditch these files. This documentation file, MENUSPUD.WRI, can go anywhere you prefer, too.

How to use VB Menu Potato to copy menus

The Menu Potato's tools are easy to understand. The window is fairly simple.

First thing to notice: You can't do anything if you haven't selected the file(s) you need to act on.

The buttons are context-sensitive. If you have chosen no source file, the buttons will all be disabled. If you have chosen no target file, the To Menu button will be disabled. So first select your forms (or files).

The window has two sides: The source side, the target side. In between, a column of buttons that executes the primary functions of the program. To copy a menu, select a source form (choosing between VB .frm files, .txt text files and .mnu menu text files) and a target form (always a VB .frm file), and click the To Form button. (How do you like the fancy picture?) The menu from the source form will be copied into the target form. And a backup copy of the target form will be made before copying the menu into it, too, just for safety. *Voilà!* Could it *get* any easier? Has it ever been any easier?

Naturally, there *are* some qualifications. You don't have to read any further than the italicized sentences here, unless you're partial to fussy details.

- 1) *Both forms must have been saved as text when originally saved in Visual Basic.* Don't worry about remembering that they must be text files; the Menu Potato will know if either or both is binary instead of text, and will tell you so. (.mnu files, which are created by the Menu Potato as described below, and .txt files are presumed to be text.)
- 2) *There must be a menu in the source form, and no menu in the target form.* The Menu Potato will alert you if either file fails in this respect, too. The Menu Potato will not copy a menu into a form file that already has a menu. (I've thought of creating a "Replace Menu" function that would first delete the current menu in the target form and replace it with the source form's menu, but wasn't sure if anyone would ever need such a thing. What do you think? Let me know if you have an opinion or a need for such a function.)
- 3) *The target form must not be in use by Visual Basic when you use the Menu Potato.* There is no chance of corruption here; the Menu Potato simply won't copy a menu into a form that is open in Visual Basic when you try to copy the menu into it. This is precisely to prevent corruption of a file. Unfortunately, there is no way to safely copy a menu into an active form within Visual Basic or while it is open. (Workaround: If you need to transfer a menu into a form you're working on in VB at the time, there is a roundabout way to do it: Remove the form from the project, copy the desired menu into it, and add it back to the project. It's a hassle, but it is generally much less a hassle than recreating a menu manually.) And, just in case you thought of it, closing the Project window won't enable VB Menu Potato to copy into a file in the open project. It just won't do it. It's for the best, believe me. The corruption that occurs when a menu is copied into a file already in use doesn't irrevocably ruin the file, and there would be a backup in any case, but it does take a lot of manual labor to repair a file corrupted in this manner. Note: I also realize that there is the chance that two projects may have files with the same names in different directories. Even if the target file is not open while a file of the same DOS filename is open, VB Menu Potato won't copy the menu into the target file. According to experts more knowledgeable than I, there is no documented way to discover what fully-qualified filenames, paths included, are actually in use by Visual Basic; what VB Menu Potato does is look for the target file's raw filename itself in the Visual Basic project window; if it finds a listing matching this name, it will not copy a menu into the target file. (You do *not* have to be running Visual Basic, or even if you are running VB, you do *not* have to have the project window open, to run Menu Potato or for Menu Potato to perform this security check.) If there *is* a way to safely copy a menu from one form into an open form, I'd sure like to hear about it.
- 4) *Menu Potato does not copy menu event code to the target form.* I suppose this would be possible, but it would greatly complicate things. I think I'll leave it for version 2. Or maybe version 3...

What else VB Menu Potato does

There are three more large buttons in the middle of the window. What they do is pretty obvious:

They extract the menu from the source file and copy it to the Windows Clipboard, a programmatically-launched copy of Notepad, or a file with the same path and name (first up-to-eight characters) as the source file and the extension .mnu. These actions allow you to manually edit menus or save them for later use without having them in a form.

Please note that when you do your own manual editing of menus outside VB, there is, outside of rigorous care, no way of assuring that your code will work in VB. I mention this because you could add anything--duplicate menu control names, bum characters, the lyrics to *Inna Gadda Davida*--to the middle of a Menu Potato-created .mnu file, and Menu Potato will still happily copy the entire text into a target form if you tell it to. But then, we've already discussed how I disclaim all responsibility for such actions, haven't we? We haven't? Well, I do.

There is just one more thing that you really need to know about VB Menu Potato's functionality. Next to the labels above the file list boxes in each column is a small button with a Notepad icon on it. I bet you can guess what it does... No, it doesn't miniaturize the Notepad executable file! It loads the file selected in the list box below it into Notepad for manual editing or viewing--provided, of course, that the file is not too large for Notepad to handle, in which case it gives you the option of opening the file in Write. (The selected filename, if any file is selected, shows in the label above the list box.)

And, of course, the About... button shows a small descriptive message box with a copyright notice, while the Exit button shuts down the program.

Okay, how much? how much?

All this for FREE? Well . . . no. Though you do get it all; I don't believe in crippling my programs.

I'd like you to register the Menu Potato if you find that you use it. Let your conscience be your guide as to how often you have to use it to "really" be using it; if it saves you just two or three times in a year from having to recreate a menu, it's probably saved you at least an hour of work. I don't ask much: \$4.95, in a check. You're a programmer, you know how important it is to see some reward for your labors of labor and love. There's a coupon at the end of this file if you feel the need to fill something out. A check alone is fine. If you want, I'll send you a laser-printed license sheet--it'll be no big deal, and you won't find any codes to unlock secret features. (I mean, this is simple stuff, right?) So if you want it, let me know. If you don't need it, we'll say your cancelled check and your good feelings about supporting shareware are your license, and we'll save a fraction of a tree.

Upgrades: I'll probably be making some changes on this thing sometime; I think it's bug- and error-free, but you may find something that doesn't quite work; just let me know and I'll see if I can fix it. But I'm not going to get into shipping upgrades; I do this stuff at night, see. You can use the latest version with no additional charge, and I'll keep the asking price low, so I don't have to ship out stuff to the probable hundreds of thousands of programmers who'll be registering this. (Yes, that's in my dreams.)

Hey, are you making some kind of VB utility that would be improved by the addition of something like this little message spud? If you want to incorporate it, let me know and we'll see if we can work something out.

For comments, compliments, problems or suggestions, you can reach me on CompuServe at address 72122,03562, on America Online at Doc Yeah or by Internet at docyeah@aol.com or dstewart@ip.portal.com. Hope you enjoy and benefit from this program.

Oh, yeah. Why Potato? It's a long story with not a great deal of meaning or significance. I came

up with Code Potato for a utility I'm still working on, one day. The Message Potato is already out there on the market. I liked the word, and that's about all there is to it. Everyone uses big, high-fallutin' names. I wanted to use something different.

Version 1.1

What's "new" in this? Well, nothing is "new." Version 1.1 fixes a nasty bug that slipped into version 1.0.1 that caused the menu to be copied into every subroutine of the target form. Quite a bug. I have no idea how it got there; must have been some other programmer on the team.

Hey, I'm sure you already knew it, but Visual Basic is a registered trademark of Microsoft Corporation. So's Windows. For that matter, so is "Microsoft." Weird. Truly weird.

COUPON!

You can send this in with your registration check if you like. If you don't, and you don't say something about it, I'm going to assume you don't need a printed license, okay?

Mail it to...

David Stewart

738 Hathaway Drive

Auburn Hills MI 48326

Program name: VB Menu Potato v. 1.1

Registration fee: \$4.95

NAME _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

☐ Send a printed license.

☐ Never mind the license; save a fraction of a tree.

Where did you find this program? _____

Any comments?