

VBToolBox © 1992

Welcome to the *Visual Basic Toolbox*:

VBToolBox is a small toolbox that resides in the lower right corner of the Visual Basic screen, always available during your session. Its very easy to spend long stretches of time switching between [break] and [run] mode fine tuning your program. If you forget to return to design mode and save your work occasionally and your system goes down you've lost everything. This program was written out of sheer frustration after loosing hours of coding work in Visual Basic because of a "UAE".

If you have any comments or suggestions for future versions you can reach me on America Online or GEnie:

America Online: HalJ 1

GEnie: H.Jurcik1

Compuserve: 71042,1566

I believe you will find VBToolBox **INDISPENSABLE!**

You may use this program for a 21 day trial period. If you agree and decide to keep using it, please register and encourage future development Only \$ 10.00

Make Checks Payable to:
Hal Jurcik
7819 S Vanport Ave.
Whittier, CA. 90606

Upon registration you will receive:

A registration # that will disable the shareware screen.
Future updates will be FREE to registered users.

DISCLAIMER:

VBToolBox is sold "as is" and without warranties as to performance of merchant ability or any other warranties whether expressed or implied. Because of the various hardware and software environments into which VBToolBox may be put, no warranty of fitness for a particular purpose is offered.

Installation:

VB-TB.WRI	This File
VBTBOX1B.EXE	VBToolBox executable
TOOLBOX1.CFG	Configuration file
THREED.VBX	System file
SS3D2.VBX	System file
HOTKEY.DLL	System file
_README.TXT	Readme 1st

VENDOR.DOC

Vendor information

Copy the files THREED.VBX, SS3D2.VBX, and HOTKEY.DLL into your \Windows\System directory. Copy the files VBTBOX1B.EXE and TOOLBOX1.CFG to your Visual Basic directory. If the file VBTBOX1B.EXE is in your VB directory it will call Visual Basic upon start up.

VB ToolBox Functions:

Save & Run:

The most important function in VBToolBox is "Save & Run". In [design] or [break] mode when you press "Save & Run" VBToolBox will save all your work and place Visual Basic in [run] mode. Now, if your system crashes everything has been saved. I have included a Global HotKey {Shift + F9} for this function to make it easy and convenient to use regularly. Save & Run should be used in place of the F5 or Shift F5 commands. Guaranteed to save you hours of lost work and frustration!!!

Note:

If you haven't given your project a name Save & Run will call the Save As dialog box.

Save/Load text

As you design programs in Visual Basic, you will normally create controls, variables, and procedures during that design process that you will later delete. Visual Basic retains these unused variables and procedures. When you create the final exe file these unused pieces of code are also added, increasing its final size. On small programs this will not have much consequence, but on larger projects where many changes have been made over time, the added size of the final exe can lengthen the load time of your programs, and lengthen download times if you distribute your programs over BBS's. In order to remove these extra definitions from your final exe's, you need to save all your event procedures on each form as a text file and then reload each one into its appropriate form. Then you must re-save your entire project before making a new exe file. VBToolBox automates the process of Save text/Load text and re-saving your project into this one command button. This process makes many API and Sendkey calls. On large projects this can take a little time so be patient. I've been using this procedure for quite a while on my machine and it has worked flawlessly, but, as always, please test this on a few of your own unimportant projects before relying on it.

Note:

If the Save text / Load text procedure ever stops and you get a message regarding the Global.bas file: Don't Worry! Visual Basic has a quirk with respect to the global.bas file. This file is never created until you actually enter code into the module, or change the name from Global.Bas to "MyProg.bas" etc. Since the default project.mak file lists a global.bas the program will call the Save as Dialog Box and ask if you wish to save this file. Go ahead and save the file global.bas, change the name if you wish, (this explicitly creates the file for that project) and then resave the entire project. Then re-run Save text / Load text. This is just a quirk in Visual basic, not an error in this program, and nothing to worry about.

Call Help:

Visual Basic's dedicated help file is called easily by pressing F1. Unfortunately if you have purchased any of the popular VB add-ons that include their own help files you must assign them to icons and return to the program manager to call them. By adding these files to the ToolBox1.cfg configuration file you can access them from within Visual Basic. To use Call Help: Click on the help button [Question Mark Icon], a list box will appear on the left side. To choose from the list double click on your choice. To remove the list dialog click with your right button,

Call Text:

If you have programming examples, read.me files for third party custom controls, global constants and API function references that you use during your programming sessions, you can add these to the [Text File] section of the Toolbox1.cfg configuration file, and access them during your VB sessions. To Use Call Text: Click on the Text Button.[File Card Icon], a list box will appear on the left side. To choose from the list double click on any title, to remove the list dialog click with the right button.

Call Paint Program:

Calls Paintbrush or any other paint program you wish from Visual Basic.

Call Icon Editor:

Calls your favorite Icon editor from within Visual Basic.

Configuring VBToolBox:

The configuration file Toolbox1.cfg is a structured sequential file. You must use a text editor such as Notepad to make any changes to this file. I have include a call for this file using Notepad in the default configuration . Therefore you may edit your preferences from within VBToolBox. If you are unfamiliar with the term structured sequential file, you may think of it as a set of Quick Basic's DATA statements with headings separating data for each function, and commas separating each variable. Do Not Change the headings or blank lines between function groups. VBToolBox uses these as keys in finding the correct variables. Under the Text File and Help File headings you will find a sub heading that contains a format example. After you finish editing this file you must restart VBToolBox for any changes to take place.

Below the headings [Icon Editor] and [Paint Program] enter the full path of the Paint and Icon Editor programs you wish to use. If you wish to substitute another program feel free. VBToolBox uses this variable in a shell call to execute the desired program.

[Not Registered]

[Icon Editor]

C:\windows\iconwrks.exe

[Paint Program]

C:\windows\pbrush.exe

Under the heading [Text Files], enter the variables for all text files you wish to access from within VB. The sub-heading gives you the format for each line.

Title\$ = Program name that will be entered into the VBToolBox list box.

Viewer\$ = Program used to view this text file, (Notepad, Write, etc.)

Location\$ = Location of text file.

DON'T FORGET THE COMMAS Title\$, Viewer\$, Location\$

[Text Files]

[Title\$,	Viewer\$,	Location\$]
VBpro Setup Kit,	C:\windows\notepad,	C:\vb\setupkit\readme.txt
VBpro HugeArray,	C:\windows\notepad,	C:\vb\samples\hugearray\readme.txt
API Cardfile,	C:\windows\cardfile.exe,	C:\windows\wrkit\Ffunction.crd
VB Constants,	C:\windows\notepad,	C:\vb\texts\constant.txt
VBpro Constants,	C:\windows\notepad,	C:\vb\texts\Const2.txt
Configure VBToolBox,	C:\windows\notepad,	c:\vb\toolbox1.cfg
VBpro Win API,	C:\dos\edit.com,	C:\vb\winapi\winapi.txt

VB Tips,

C:\vb\texts\VB-Tips.exe,

Under the heading [Help Files] you will also find a sub-heading outlining the format of this section:

Title\$ = Title of help file to be entered into the VBToolBox selection list box.

HelpFile\$ = Path and name of Help file.

If your help files are located in your Windows\System directory all you need is the file name.

This function will only launch help files. Files that don't use the WinHelp.exe help engine should be entered in the Text File section.

[Help Files]

[Title\$, HelpFile\$]

VB Knowledge, C:\vb\vbknowlg.hlp

Control XRef, C:\vb\ctrlref.hlp

Setup Kit, C:\vb\setupkit\setupkit.hlp

Widgets #1, SS3d.hlp

Widgets #2, SS3d2.hlp

Widgets #3, SS3d3.hlp

Win Api, C:\vb\winapi\winsdk.hlp

API XRef, C:\vb\winapi\apixref.hlp

*** End Config, VBToolbox ***

Important:

After you finish making changes to the Toolbox1.cfg file you must exit and restart VBToolbox for the changes to take effect!

ONE LAST NOTE:

- 1. DON'T CHANGE THE HEADINGS OR SUB-HEADINGS, VBTOOLBOX USES THESE AS KEYS TO INSTALL THE CORRECT VARIABLES IN EACH FUNCTION.**
- 2. LEAVE A BLANK LINE BETWEEN EACH FUNCTION GROUP.**
- 3. YOU MAY ENTER AS MANY TEXT FILES OR HELP FILES AS YOU WISH UNDER THE RESPECTIVE SECTIONS, BUT THE ICON EDITOR AND PAINT PROGRAM FUNCTIONS ONLY TAKE ONE ARGUMENT VARIABLE!**
- 4. DON'T FORGET THE COMMAS SEPARATING THE VARIABLES!!**

Correcting Errors:

Since VBToolBox is intended for programmers in the Visual Basic environment I assume this configuration file example is fairly easy to follow. Just substitute your file names and locations for the default examples and everything should work fine. (Famous Last Words!!). If I'm wrong send me some nasty E-mail on GEnie or America Online and I'll add a configuration dialog in the next version.

In the Help File Function if you get the big blue error message check your entry for proper spelling and the correct path.

In the Text File Function, if you receive an error message from the text file viewer that you selected, check your Location\$ entry for proper spelling and path. If you receive a load error message from VBToolBox, check your Viewer\$ entry for spelling, path errors and a trailing comma. Check your Title\$ entry, it must be 20 characters or less and must be followed by a comma.

If the number of entries in your Text File list box or Help File list box don't match the number you have entered into the configuration file YOU FORGOT A COMMA, OR YOU ADDED A BLANK LINE BETWEEN ENTRIES. All variables must be separated by a comma and all function groups must be separated by a blank line. No blank lines are allowed between sets of variables, only between function groups. Follow the default example...

If you have any comments or suggestions for future versions you can reach me on America Online or GENie: Or by writing to me at the address below.

America Online: HalJ 1

GENie: H.Jurcik1

I believe you will find VBToolBox INDISPENSABLE!

You may use this program for a 21 day trial period. If you agree and decide to keep using it, please register and encourage future development Only \$ 10.00

Make Checks Payable to:

Hal Jurcik
7819 S Vanport Ave.
Whittier, CA. 90606