Visual Basic ToolBox Pro

 CIS ID: 71042,1566

Take a Quick Tour

Visual Basic ToolBox Pro extends the native capabilities of the Visual Basic programming environment, maximizing your productivity by giving you instant access to all your third party add-on packages, combining functions to maximize the power of each mouse click, and adding features that save time and frustration enabling you to concentrate on your code, not manipulating the programming environment.

This shareware version is a scaled down or Lite version of ToolBox Pro intended for evaluation purposes ONLY! You my use this program for 30 days to give it a thorough run through before purchasing the registered version. I'm sure you'll find ToolBox Pro an indispensable addition to your VB programming environment.

Detailed Operations and Functions

Ordering Info
Installation
Configuration

ToolBar
Code Manager
hWnd & VB Spy
Compression Manager
PicViewer

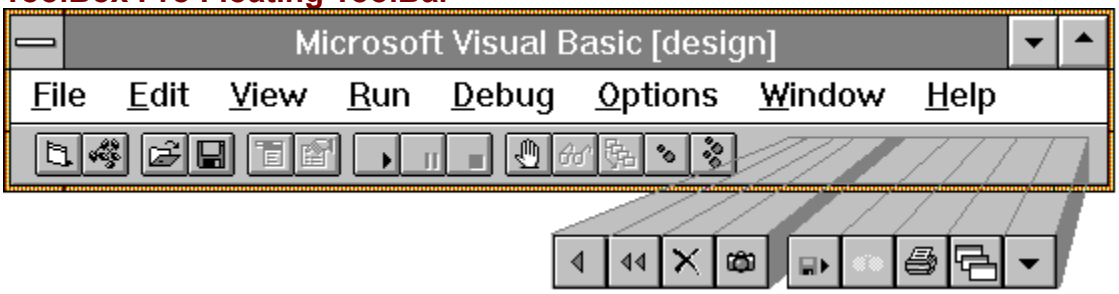
Save & Run
Saving Code as text files
Printing Code
Call Text Files
Call Help
Launch Programs
Vbx Files
Program Hotkeys
PopUp Menus

Program Suport

Quick Tour

Here you'll find the major sections of ToolBox Pro. Click on any button and get a detailed description of its function and operation.

ToolBox Pro Floating ToolBar



Main ToolBox

Program Setup and Tutorial
Installation
Ordering Info
Program Suport

Program Setup and Tutorial:

Overview:

ToolBox Pro began as a humble attempt on my part to add "Wish List" functions to the Visual Basic environment making my programming sessions more productive and enjoyable. As the program has grown, so has the complexity, daunting many new users. In this section I will attempt to explain my personal way of configuring and using this program to get you up and running. By no means is this the only way of using ToolBox Pro, but it should give you a base for experimentation. Once you become comfortable with the program features you can go back and alter the configuration to your preferences.

Make sure ToolBox Pro and Visual Basic are both running. Now Minimize the main help window, and use this small tutorial to walk you through the configuration process, and a brief overview of this new addition to your programming environment.

Press the "Next" button to go on to the Installation and Setup section

Installation and Setup Configuration:

If the configuration dialog is not open press the text button and double click on the "Configure ToolBox Pro" entry to call it up and lets get started.

Set your VB Location:

Before we can discuss anything about day to day usage, you must have the program setup for your system and working properly. The first entry in the configuration dialog is the location of your VB directory, and this should be the first entry you confirm. One of the main features of this program is the ability to launch your complete VB working environment in one operation. When you double click on the ToolBox Pro icon, if Visual Basic is not running on the system, the programs first task is to look for the VB.exe file and start Visual Basic for you.

Current Project:

The seventh entry in the configuration dialog is labeled "Current Project". Even though many programmers work on multiple projects, most people work on one major project at a time, coming back to that project every morning. Choose the project you wish loaded when you first start VB in the morning. Enter the full path in the text box labeled "Current Project". First, click on the "Current Project" text box with the mouse, giving it the focus. Now press F10. This will call up the browse dialog box. Use the file controls to find your current project *.mak file and double click on that file in the file list box. It will automatically be entered for you in the text box, avoiding typing mistakes. Once configured, when you start ToolBox Pro, it will find the vb.exe file and start Visual Basic. Then it will load your current project. Then it will finish loading ToolBox Pro. Again, a double click on the ToolBox Pro icon will load your complete working environment in one operation.

Launch at Startup / Launch State:

This option will not be used by everyone, but its worth mentioning early so you know its function. In addition to launching VB and your current project, at startup, ToolBox Pro is capable of launching one other program. If you have a third party database add on, other VB related program, or if your doing a lot of DDE programming in your project and you need a particular program running in the system for the DDE connections to work. Enter that program here and it will become part of your startup working environment.

For demonstration purposes lets enter the Windows Paint Brush application for our startup option. First click on the text box labeled "Launch at Startup", giving it focus. Press "F10" to call up the browse dialog box. Use the file controls to go to the Windows directory and find the paint brush exe file; "pbrush.exe". Once you find it, double click on the file in the list box to make the entry in the configuration. The very next text box is labeled "Launch State". Enter a "1" if you wish to start this program in its normal state, a "2" if you wish to start this program in an Iconized state. For this

demonstration enter "2"

Lets Review . . .

As of this point you have entered the correct path to your Visual Basic directory, listed the current project your working on, and listed a second program you want ToolBox Pro to launch at start up. This is your working programming environment. If VB is **Not** running when you launch ToolBox Pro, your working environment will be launched at startup. If ToolBox Pro detects a running version of Visual Basic it will skip this process and load only itself.

Note:

If you do not wish ToolBox Pro to load any programs or projects on start up other than VB leave these items blank.

Press the "Next" button to go on to the Setup Preferences section

Setup Preferences: « Back | Next »

VBx Location:

Enter the location of your design time VBx files. The default is your C:\Windows\System directory. This enables the "VBx" button in the ToolBox and gathers a list of all available VBx files in your system. When you click on that button a list of your available VBx files will appear. Double click on your choice and it will be added to your project.

User Configurable Buttons:

ToolBox Pro allows you the user to configure three buttons for your use. I suggest for now you stay with the defaults. The left arrow button in the ToolBox will launch the Windows File Manager. The two left ToolBar buttons are configured for menu commands. The first one is set for the menu Edit Paste command, the second calls up the procedures list. Once your familiar with the program you can come back and make changes.

Printer Designation:

This one is simple. Choose your printer type. Standard values will be set for you that should be correct. You can always come back and change these values to fine tune your printer output if you wish. Detailed instructions can be found under the Configuration section.

Tbar Location:

If your using 640 x 480 resolution this number should be "4". If your using 800 x 600 or 1024 x 768 this number should be "1". This sets the ToolBar position in relation to the VB ToolBar. Set it and FORGET IT!!

Press the "Next" button to go on to the Launch List Configuration section

Launch List Configuration: 

You are finished with the "Startup" section, now press the "Launch" button.

This section is easy, in fact its self configuring. In the upper left corner there are two option buttons. **Config Program Launch List** and **Config Help File List**. First click on the Config Program Launch List option, and then press the "Auto Setup" button. This will cycle through your windows directories and your VB directories and create a basic working configuration for the Program Launch List. After this process is complete choose the Config Help File List option and press the "Auto Setup" button again. This will cycle through your VB directories and add all your help files to the Help File Launch list. If ToolBox Pro finds any help files it can't identify, an input box will appear with the file name and ask if you wish to add this file to the list, or pass it up. If you wish to add this file, give it a name in the text box provided. If not, press cancel to skip that file.

Your almost finished!!!

Your initial configuration is complete, all you have to do is press the "RE-Config" button to write the new information to the configuration file.

Now to test your new configuration exit Visual Basic, ToolBox Pro should shut down also. Now double click on the ToolBox Pro Icon. Visual Basic should start, along with your current project if you made that entry and possibly a second program, and finally ToolBox Pro will appear.

When if everything loads correctly press the next button where you can start exploring the program features.

Exploring the Program: « Back | Next »

Now that you have a basic configuration setup its time to take a quick run through. If you haven't already, start ToolBox Pro and let it load your programming environment. Once everything is loaded you should see a second ToolBar to the right of the default VB ToolBar, and a ToolBox in the lower right corner of your screen. I'm assuming your favorite project is loaded and ready to go.

Before we start, click on the VB title bar caption to give Visual Basic the focus. Now minimize the ToolBox Pro toolbar. Toolbox Pro continuously monitors your environment, inserting the toolbar when you are working on code, removing the toolbar when a help file or other program has the focus. Since you will be switching focus between this help file tutorial and VB, this process can be distracting.

First:
The first function we'll explore is the **Create/Print txt files** function. This is the main backup and indexing function in ToolBox Pro. You should run this process with every project you are working on, whenever you make any significant changes. When you run the Create/Print txt file function ToolBox Pro cycles through your *.mak file, creating a backup set of ASCII text files of every form and module in the project. It also indexes these files for use in the code manager and compression manager.

Lets get started:

Press the tools button  in the lower ToolBox. A list of four buttons will appear to the left of the that ToolBox. Now press the bottom button labeled "Create/Print txt files". A dialog box will appear that will allow you to save your current project as a series of ASCII texts files, print them if you wish, and catalog and index this project in the code manager. When beginning a new project I always create a separate directory to contain all of the new project files, and I always store my text files in that project directory. If you don't adhere to this practice already, now is a good time to start. Use the file directory to choose your current project directory and press the "Create txt Files" button. Depending on the size of your project this may take anywhere from a few seconds to a minute or more for large projects.

This function will cycle through your currently open project, create a text file for each form and module, enter that file into a list box, and finally index each file and its related sub routines and functions in the code manager for use at a later time. When the process is finished you will see a list box containing each file created at the bottom of the screen.

Print Demo:

For demonstration purposes, make sure your printer is on. Choose one of the files

with your mouse. Now click on the list with the right mouse button and press the portrait button that appears. The file will print along with a header containing the project name, date, form name and time.

Review:

So far you have created a basic configuration for the program and used the Create/Print txt file function. If you've gotten this far everything is setup correctly and your well on your way.

Now lets review how and why to use this function.

1. First and foremost creating a set of ASCII text files for your project is a great backup procedure. I run this process before I make any major changes to a project. If I change my mind, or the changes I've instituted don't work to my satisfaction I can always use the load text function to replace the old code and return to the original working code,... before any changes were made.

2. Viewing code on screen is very straining on the eyes, especially running 640 resolution on a 14" monitor. If I want to review code, its easier to print out the project and go through it at my leisure in a comfortable chair instead of hunched over the monitor.

3. Everytime you run this process for a project, all text files are updated to your current code, the project is cataloged in for use in the code manager and the compression manager.

Press the "Next" button and lets explore the Compression Manager

Compression Manager: « Back Next »

The compression manager has two functions. It will search through the the project *.mak file, and the associated ASCII text files and list any VBx 's and DLL's that you have used, adding them to a compression list. Once your list is complete, the compression manager will handle all the details of compressing the files for your installation program. Lets give it a run through.

Press the tools button  in the ToolBox again, but this time choose the button labeled "Compress for setup". When the compression dialog appears use the file controls to find and choose the *.mak file of the project you currently have loaded. The same project you used in the "Create/Print txt file" function.
This is the automated part of the compression manager. If you choose a *.mak file, the program automatically searches that mak file and the associated text files in that directory, generating a compression list of all DLL's, VBx's, the exe, and the runtime files needed for distribution. Once the list is created, feel free to add or delete any file you wish. When your satisfied, press the "Compress" button. A sub directory will be created below your project directory named "Comp", and the files in the list will be compressed and stored there.

Of course if you have any help, text, data or ini files you must add these to the list yourself. You may also use the compression manager manually to select files for compression. Just select the individual files using the file controls instead of the project *.mak file.

If you have VB 2.0 this is the perfect way to prepare your files for your installation program. If your a VB 3.0 user, the new program comes with a setup wizard, but when you use the setup wizard you must recreate the entire setup program everytime you have a change. The compression manager is the perfect way to compress one or two files as they are changed and add them to your setup program without having to re create the whole setup process through the wizard each time.

Press the "Next" button and explore the Code Manager

Code Manager: « Back Next »

Overview:
The code manager is a central storehouse of reusable code that you can cut and paste into existing and future projects. The key to its ease of use is the **Create/Print txt file** function, and the code managers ability to import and export module files.

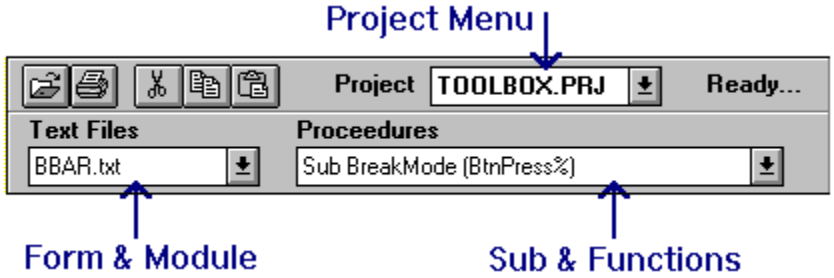
The code manager comes preloaded with the Win 3.1 API declarations, Type definitions, Global constants along with the Visual Basic Global Constants and Data access Constants. If you have any third party add-ons that require the use of their own set of constants and declarations you can easily convert these files for use through the code manager also.

Now lets take a brief walk through the code manager and look at its basic features.

There are two ways to access the code manager. One is through the tools button  that your already familiar with, and the second is the code manager button on the ToolBar

 . Lets load the Code Manager now.

The first thing you'll notice is the name of your project in the top combo box. In the bottom left combo box will be a list of the text files that were created for each form and module. In the bottom right combo box you will see a listing of every sub and function in your project.



Lets continue our walk through and demonstrate some of the features of the code manager. In the bottom right combo box marked procedures, drop down the list and choose one of the subs or functions. Upon selection, the code manager searches through the appropriate text file, and loads the selected code procedure in a window.

1. In the edit menu choose "Select All" to highlight all the code in the window.
2. Next click with the right mouse button anywhere in that code window to call up the PopUp command menu.
3. Double click on Copy in the PopUp menu
4. Click on the ToolBox Clipboard Window to give it focus and then click with the right

mouse to call up the PopUp command menu again

5. Double click on Paste in the PopUp menu to paste the code into the ToolBox Clipboard.
6. Repeat this process with one or two other procedures to build up a sample file in the ToolBox Clipboard window.
7. Next in the Options menu select API manager.
8. In the Search Menu select find
9. Type in "IsZoomed" and press the "Find_Decl" Button. The declaration for the IsZoomed API function will be added to the ToolBox Clipboard.
10. Drop down the API declaration combo box and choose a few more declarations

You now have a general idea of how to access code from existing projects and build function groups for addition to your current project. Before we finish with this demo lets go through one more process.

Move the cursor to the upper left of the ToolBox Clipboard code window and press enter to give yourself a a blank line at the top of the window. Now with the left mouse button double click anywhere on the code window to call up the PopUp Code menu. Double click on the first entry "Project Heading". This will insert a standard Project heading at the beginning of the code window. Now in the file menu select "Save as *.lib" to save this file as a library file for future use.

Lets Review:

In the code manager you have called up a couple of procedures from one of your projects, cut and pasted the various subroutines into the general clipboard, added a few API declarations, and saved the result as a library file for future use.

While we have a few code windows open lets go through a few more features. Click on the code window you have just created giving it focus. Now press the print button in the Code Manager ToolBar. All code will be selected and sent to the printer with the window name, time and date.

Now create a new project, give it a name and save it to its own directory. In the edit menu select "Export Library as Module". Double click on the library file you created from your last project. A new module will be created in your new project. If you haven't given your new project a name, the code manager will prompt you with the "Save As" dialog before it will let you add any new modules.

Press the "Next" button for 10 tips on utilizing and maximizing your new programming environment.

Maximizing Tips: 

Now that you have completed a general run through of ToolBox Pro's features and capabilities here are a few tips for maximizing this program.

1. Load each project currently on your system and run the "Create/Print txt file" function. This will give you an ASCII text file backup, and catalog the project in the code manager so that you may have instant access to the code during your future programming sessions.
2. In the Edit menu use the "Import Module as Library" function to convert your existing project module files to code manager libraries. Once imported to the code manager you can convert these files to reusable standard libraries of functions that can then be exported to future projects using the "Export as Module" function.
3. Re-run the "Create/Print txt file" routine before and after you make any major changes to a project. In this way you will always have a backup to return to, and your code manager files will always contain the latest code information.
4. Start using a descriptive naming convention for all your controls and sub routines. When trying to find that piece of code from an old project, "ReConfig_Click" is much more telling than "Command3_Click"
5. Get in the habit of using the Save & Run button or pressing "Ctrl + F5" when running your projects in the VB environment. This works in design and break mode, and saves your project BEFORE entering run mode. If your system crashes while your testing your program, no code will be lost. For VB 3.0 users you might want to disable the Save First setting in the VB .3.0 Environment menu. All it does it inundate you with reminder messages, Save & Run saves automatically. And you can always use F5 to run a program without saving.
6. SET and USE the AutoSave timer. With the combination of the Save & Run button and hotkey, and the AutoSave timer you have no excuse if your loosing work because of a GPF.
7. Once your familiar with the program configure the PopUp Command Menu with your most used menu functions. The PopUp menu gives you instant access to your menu functions with a right click of the mouse on any code window. It's not quite drag and drop editing, but its close.
8. If you use standard constants or variable names in most of your projects, add them to the PopUp Code Menu. Both PopUp menus are user configurable and can save time and typing errors.
9. There is no reason to leave the Visual Basic environment to hunt for other programs or help files. Configure the two launch list boxes with programs you access while in Visual Basic, and all VB related help files. Once you've configured the launch lists you can leave the program manager minimized and concentrate on your work.
10. Keep you code modules well below 32k to avoid mysterious out of memory messages. Organize your modules keeping all related code together i.e. ...

Printer.bas, System.bas, File_IO.bas etc. Module code is loaded as needed. Putting a little thought into structure and usage at design time will cut down on resource usage, and make your programs work faster for the end user. ToolBox Pro has a resource monitor to help keep track of your programs resource usage. Use it to monitor memory usage as you load, hide, show and unload forms.

Thanks for trying ToolBox Pro. I'm sure you'll find this new working environment indispensable.

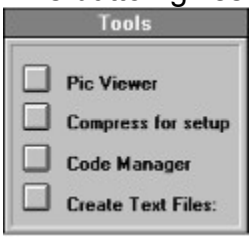
ToolBox Save & Run

Pressing this button in design or break mode will first save and then run your project.
Hotkey for this function = CTRL + F5

For more information on this function see
Save & Run

ToolBox Tools Menu

This button gives you access to the ToolBox Pro accessories



For more information on these function see

PicViewer

An *.ico, *.bmp, *.wmf image viewer

Compression Manager

Searches your project for VBx and DLL files and handles compression

Code Manager

The ultimate code management utility completely user configurable

Saving Code as Text Files

Contains Backup, Printing and cataloging routines.

ToolBox VBx List

List of all available VBx files in your system
Double click on any entry and instantly add
to your project.

For more information on this function see
Vbx Files

Help File List

List of all third party help files
relating to Visual Basic.
Completely user configurable

For more information on this function see
Call Help
Configuration

ToolBox Pro Spy Window

Using the cursor as a pointer this window gives a continuous reading of the Handles, Classname, Caption, Parent Window, and Program for any open window or control running in your system. VB or Windows

For more information on this function see
hWnd & VB Spy

Launch Button

This button is user configurable to launch any program you wish. Perfect for instant access to a paint or icon program or even the file manager!

For more information on this function see Launch Button Configuration
Configuration

Program Launch List

This button calls up a program launch list.
All entries are completely user configurable.
This feature allows you access to any program
without leaving the Visual Basic environment.

For more information on this function see

Launch ListBox Option
Configuration

Text file / Configuration List

This button calls up a list box with two functions.
1. Gives you instant access to constant, readme text files for your VB add-ons using the default text editor.

2. At the end of this list box you'll find entries for the ToolBox Pro configuration files.

For more information on this function see

Text Listbox Option Configuration

Exit

You can use this button if you want to close ToolBox Pro without leaving Visual Basic. If you exit VB, ToolBox Pro will shut down automatically.

Create txt / Print txt Overview

1. This function will cycle through your current *.mak file creating a backup ASCII text file for each form and module in your project. As each text file is created the file name is added to the list box to the left of the toolbox. This process also catalogs each sub and function for instant access using the code manager.

2. There is no default directory specified for your text files. To prevent future organizational problems it's best to create a separate directory for each project you are developing and create your text files in that directory.

3. Once this process is complete, ToolBox Pro can print any or all files just created for your current project. Using the mouse you can choose any or all files from the list box, a right mouse click on the list box will send the selected files to the printer. If no files are selected a right mouse click on the list box will close this function.

NOTE:
This operation will overwrite all previously created text files for this project. If you wish to save an earlier set of text files, save them to a new directory or floppy.

Program Support

For all Registered users:

If you have any problems, questions, suggestions for future features, or bug reports you can contact me through the following channels

COMPUSERVE:
Ci$ ID # 71042,1566

Fax# 310-908-1054

Mail:

Hal Jurcik / HalDen Studios
7819 Vanport Ave
Whittier, CA. 90606

Phone:
Mon. - Fri. 9 am to 5 pm Pacific time
310-695-5228
I run a small shop, therefore I'm in and out of the office throughout the day. If you need an answer by phone its best to call around 9 am or just before 5 pm.

Using the Compression Manager

The compression manager is a quick and efficient way to locate and compress all files in your project in preparation for developing an installation program. The compress program that is included in the VB professional edition is a DOS program, used from the command line and is a pain to use for most of us who have become accustomed to the Windows environment, hence many people don't use it. ToolBox Compression Manager is a Windows front end for Compress.exe that handles all the details for you. In addition its searches your project files for VBx's and DLL's you've used in your program and adds them to a compression list. Just a little help for those of us with overused and forgetful brain cells!!

For VB 3.0 users, the compression manager is an easy way to selectively compress files that have changed, and update your existing installation programs.

1. Use the file controls to find and select your project *.mak file.
2. Once selected the compression manager searches your project *.mak file for any VBx files and adds them to the compression list.

   Note: Some third party VBx files require a separate runtime version to be distributed with your program. Your *.mak file contains the location of the design time versions, usually in your windows\system directory. Make sure you replace these with the runtime VBx's before you compress the group.

3. If you have used the ToolBox Pro's save text routine, the compression manager also searches all text files associated with your project for DLL's that need to be included and adds them to the compression list also.

   [All DLL's are entered into the compression list without a path and are assumed to be in your \windows\system directory. If a particular DLL is in another location use the file controls to re enter it into the list so the path is correct.]

4. If a text file corresponding to any *.frm or *.bas file cannot be found, a "can't find" notification for that file will be added to the list. This is just a notification that this particular file does not exist in the project directory because it was created recently and has not been saved as a text file, and therefore can not be searched.

   [Compression Manager functions manipulate only the text files related to your project, never the original forms or modules. This allows you to use the binary option for saving and loading projects if you desire, and guarantees that ToolBox Pro will never make any changes to your original code files.]

5. Now use the file controls to select any *.ini, *.cfg, data files, and setup files that need to be included in your installation program.

6. The default compression destination is ..\projectfile directory\comp. If it doesn't exist it will be created, if you wish to change the destination feel free. If your new selection does not exist it will also be created.
7. When your satisfied with your final list of files press the "Compress" button and all files will be compressed to the destination directory

Viewing graphics with the PicViewer

The PicViewer is a simple graphics viewer that supports the three file formats in Visual Basic, *.wmf, *.bmp and *.ico. After a very short time most VB programmers collect mounds of small graphic images. Using your up and down arrows you can easily search directories for that perfect work of art!! :)

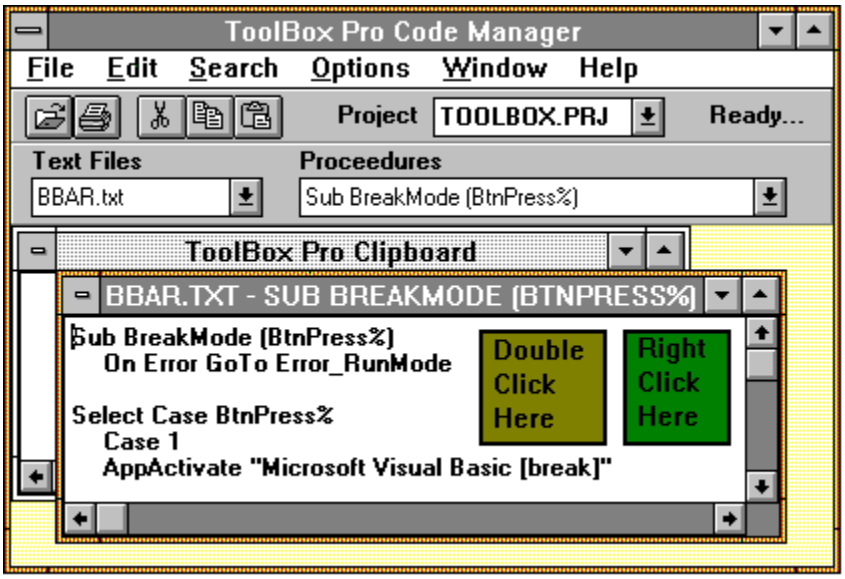
ToolBox Pro Code Manager

Feature List

The new Code Manager gives you instant access to any piece of code from any project in your system along with all Win 3.1 API function declarations, Global Constants and Type definitions. You may now cut and paste code from various projects or from your own standard library of reusable functions, without leaving your current project. Over time you can easily build generic *.bas modules of standard functions or skeleton apps for time critical projects.

This version comes pre loaded with the Win31 API Declarations, Global Constants, Type Definitions, Visual Basic Global Constants and the new access engine Data Constants. Users have the capability to convert all type definitions, global constants and DLL declarations from your third party add-ons for use by the code manager.

Click on any ToolBar button or combo list box for a detailed description of its function



Getting Started

Creating reusable Code Libraries and Modules

API Manager

File Open Button

Calls the file open dialog box.
The code manager can open any
text file, along with your *.bas and *.frm
files if you save them in text mode

Print Button

This button will print the currently selected text in the active code window. If no text is selected this function will select all text in the current window and send it to the printer.

Cut Button

This button will cut the selected text to the Windows Clipboard for pasting into any code manager code window or any Visual Basic code window

Copy Button

This button will copy the selected text to the Windows Clipboard.

Paste Button

This button will paste the contents
of the Windows clipboard to the currently
active code manager code window

Project List

In the Code Manager this list contains
all the projects available on your machine.
Everytime you use the Create Text File
function your project is automatically
indexed for the code manager

If your using the API Manager {Options Menu}
this list contains a list of available DLL declarations
type definitions, and Global Constants.

Form / Module List

In the Code Manager when you choose a project this list will contain all the text files created from each form or module.

In the API manager this list contains group headings for Global Constants, Type Definitions and DLL Declarations

Procedures List

This list contains every sub, function control for each form selected. When you make a choice from this list the code manager will open the code file search for the selected code for that sub, function, or control and add it to a code window for editing.

PopUp Code Menu List

You can call this Code Menu List by double clicking on any code manager code window or any Visual Basic code window. This menu contains reusable code snippets that can save you time and keystrokes in you work Double click on any selection and that piece of code will be inserted at the current cursor position in the working code window.

A right click on the menu will close.

In this registered version the code snippets in this menu are totally user configurable.

PopUp Menu List

A right click on any code window in the code manager or in a VB code window will call up this PopUp Menu List. Double Click on any selection to choose that command.

A right click on the menu will close

In this registered version the choice of menu commands in this popup menu is user configurable.

API Manager

One of the great features of Visual Basic is the ability to use the Windows API functions in your programs. The API Manager gives you instant access to the Windows 3.1 API declarations, Global Constants, and Type Declarations.

The code manager can be configured to work with almost any combination of third party add-on's to Visual Basic. If you have constants and DLL declarations from third party add on products you can configure those files and add them to the API manager for instant access in the future.

Again the easiest way to familiarize yourself with general feel of the program is to just jump in and experiment. You can't erase anything. API Declarations are indexed by their DLL. Just click on a selection in the combo box and that declaration will be copied to the ToolBox Clipboard. Or use the search dialog to search the combo box for a particular API call, when found it will be added to the ToolBox Clipboard. The Global Constants and Type Declarations will be displayed in a list box window. After you have made your selection a right click on the list box will also copy your selection to the ToolBox Clipboard. After you finish building your declarations section just cut and paste into your desired project form or module.

Visual Basic is "VISUAL", go experiment.

Code Manager
Getting Started
Creating reusable Code Libraries and Modules

Creating reusable Code Libraries and Modules

Using the Import / Export Functions

This version of ToolBox Pro comes pre loaded with a basic library of reusable code to get you up and running.

As you write subs and functions that may be reusable in other projects you can add these pieces of code to a library file for easy access later. Once created these files can be added individually or as a group to create a standard set of functions and a basic framework for future projects.

Try to keep related functions and declarations together, for example; all printing functions in one library, all system functions in a second, database functions in a third, screen positioning functions in a fourth. You might also wish to create a standard library bas file with general functions you wish to add to each project. Either way if your consistent from the beginning, take care in writing functions you can reuse, using these code libraries can make the development of future projects much easier.

Using the Import / Export Functions

Under the Edit menu in the code manager, you'll find functions for Importing and Exporting code modules to and from the code manager. If you save your projects in text form rather than binary, you can use the import function to convert your existing modules into a reusable library file. Once imported into the code manager you can add or modify the file into a standard add on library of functions. I recomend you create library files grouped for a particular pupose, Examples would be Printing Functions, System Functions, File IO, Screen Placement, Database Management, Sorting and manipulation, Error checking, etc... When creating the initial framework of future projects you can use the Export function to add these existing library files to new modules in these projects.

In addition to the import and export functions you will find the "Delete *.prj/*.lib" function. This will allow you to delete old project entries or libraries once they are no longer usefull.

The registerd version contains some examples of library functions as an incentive to your future organization.

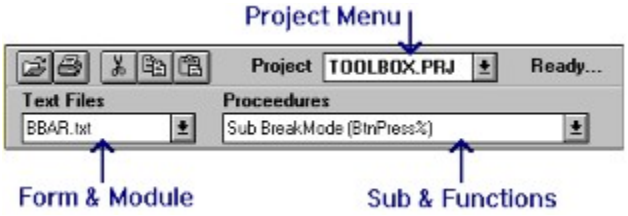
Getting Started

The new Code Manager is integrated with the ToolBox Pro Save Text function and with Visual Basic itself. Everytime you use ToolBox Pro to save your project as a series of text files, Code Manager automatically catalogs your project location and all forms and modules along with their associated sub's and functions for future access. A more self explanatory way of familiarizing yourself with the capabilities of this part of the program is to use it.

1. Open two of your projects and use ToolBox Pro to save them as text files.

2. Press the  button to access the tools functions. The last button labeled "Create/Print Txt Files" will cycle through your project and create a text file of each form and module in one operation. Repeat the process for a second project if you have one and your ready to open the Code Manager and explore its features.

3. When you open the code manager you will find the projects you just saved listed in the project menu with the extension *.PRJ. The form and module menu will list all the text files that you created. The Procedures menu will have a list of all the subs and functions within each text file.



At this point choose one of the sub's in the Sub & Function menu and experiment away. You can't hurt anything, or erase anything. Remember your working with the text files you created, not the original forms or modules. You can always recreate them.

One note on what's happening is appropriate here: When you choose a sub in the Procedures menu, Code manager opens the text file, finds the desired piece of code, copies it to its own window and closes the file. This is useful when you need to locate a specific routine in an old project and cut and paste to your current form or module. If you wish to open and edit the complete text file for a particular form, look under the file menu for the Open txt file command.

Note:
One thing you'll notice quite quickly once you start using the code manager is how important a good naming convention for controls, and descriptive names for your sub's and functions can be. Example: Command1_click tells you much less than Re_Config_Click about what type of code lies behind that control when searching for

that reconfiguration function you'd like to reuse in your current project.

Feature List

1. Automatically catalogs and indexes your project when you use the ToolBox Save text function
2. Edit any text file
3. Instant access to any piece of code from any project in your system
4. Edit project *.frm and *.bas files, if you save your project in text form.
5. Build libraries of reusable modules, API functions and constants
6. Instant access to Win 3.1 API declarations, Type Declarations, Global Constants, and VB Constants.
7. The API manager can be configured to utilize Global Constants and DLL declarations from any third party add-ons giving you the same easy access with any future product you might purchase.

PopUp Menus

ToolBox Pro has two user configurable PopUp menus:

The PopUp command menu gives you easy and instant access to menu commands of your choice. A right click on any code window will call this menu, Double click on a selection to make your choice or a right click on the menu will close. In the registered version the PopUp menu is completely configurable to your preferences.

The second PopUp code menu contains snippets of reusable code that you can instantly paste into an open code window. A double click with the left mouse button will call up this menu. Double click on any selection and the selected code will be inserted at the cursor position in the currently active code window. A right mouse click on the menu will close. Again completely user configurable.

Save & Run

The most important function in ToolBox Pro is "Save & Run". In [design] or [break] mode when you press "Save & Run" ToolBox Pro 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 {Control + F5} 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.

VB 3.0 users:

In VB 3.0 its best to disable the Save Before Run environmental setting and use the ToolBox Pro Save and Run function. The VB 3.0 setting calls a message box and prompts you to save before entering run mode creating extra steps. The ToolBox function is automatic.

In design mode ToolBox Pro has a function that will save your work at specified intervals. See the configuration section for more information on configuring the Auto Save feature.

Configuration

Call Text Files

I spend a lot of time browsing different Online services such as COMPUSERVE and AOL. There is a lot of message traffic on these services, some containing useful programming examples. When I find a message thread that contains an interesting code sample, I've always saved it for future use. Unfortunately two months later when I need that piece of code, finding it was always a problem.

To use this function you must create a sub directory under your VB directory to store all your programming examples, and third party text and constant files. I call mine "Textfile". Copy all your third party text files to this directory. When you find an interesting code snippet you wish to save, put it in this directory. In the configuration dialog, List this directory in the box labeled "Text File Location", choose a default text editor for viewing these files. Notepad should suffice unless they're very long. When ToolBox Pro loads all files in your Textfile directory will be added to the text file list box. Now that your organized you'll have easy and instant access to these files in the future through the text file list box.

To use the Call Text function Click on the Text Button.  , a list box will appear on the left side. To choose from the list double click on any title, ToolBox Pro will launch that file using the default text editor. To remove the list dialog click with the right button.

Also see:
Configuration

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 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. A right mouse click will close.

Also see:

Configuring the Help Listbox

Vbx Files

ToolBox Pro allows you to add Vbx files to your project with the click of your mouse. Click on the Vbx button and a list box will appear to the left containing all available Vbx files. Double click on any file with the left mouse button and it will be added to you project. A click on the list box with the right mouse button will close it. For this function to work properly enter the full path of the location of your Vbx files in the Vbx Location text box in the configuration dialog.

Vbx Location=C:\Windows\System

For a more detailed explanation of the configuration process also see:

Configuration

hWnd & VB Spy

In order to use the Windows API functions you must have the handle of the program you wish to manipulate from within Visual Basic. This function will find and list all active apps running in windows and give you a task ID#, handle, and caption for each. The handle list box has two options: The default shows only visible windows and their handles, the second option also includes all invisible windows with captions. Don't save these handles as constants in your program, this list is for debugging purposes only. Use only to confirm the accuracy of your code. Handles are randomly assigned when an app. starts.

The VB Spy dialog will give you the handle of any control in any active window along with the classname and caption. It will also supply the same information for that controls owner window and parent application. Just place the cursor over any control or form you want information on. Perfect for checking API calls!

Note: For an accurate reading of a controls hWnd the control should be active. [not disabled]

Configuration

The command for calling the configuration dialog is located in the text file list box, at the end of the list. Just double click on the name "Configure ToolBox Pro".

The configuration dialog contains two sections.

The Startup Section consists of a series of text boxes to enter instructions for user options. When you tab or click on a text box, basic instructions and defaults will be displayed in the information panel located in the center of the dialog box. To browse your directories for the appropriate files or programs press F10. Double click on your selection and it will be added to the configuration automatically.

The Launch Section allows you to configure the Help file launch, and Program launch functions. When you first install the program use the Auto Setup button to create a standard configuration for Help files and the Launch Program function for your system. Once familiar with the program you can return at any time and add or delete any program item

When you are finished with your changes make sure to press the Re-Config button to write the new information to the configuration file. You must exit ToolBox Pro and restart for changes to take effect.

Printer Type
Launch at startup
Button Launch Configuration
Loading Your Current Project
Configuring Toolbar Buttons #1 & #2
Printer settings
TBar Location
Auto Save
Min on start / On Top Options

Configuring the Launch / Text / and Help listbox's

Launch ListBox Option
Text Listbox Option
Help Listbox Option
Correcting Errors

Printer Type

Printer Type:
Default: Laser/Ink Jet

Select your printer type by choosing the appropriate option button:
1. Laser/Ink Jet, 2. Dot Matrix, 3. Wide Dot Matrix

The **laser/ink jet** designation enables both portrait and landscape printing capabilities and sets the default values for characters per line and lines per page.

If you have a **standard dot matrix** printer that uses 9.5" perforated paper, use the **Dot Matrix** designation. This instruction disables landscape printing.

If you have a **wide carriage dot matrix printer**, Use the **Wide Dot Matrix** designation. This instruction also disables landscape printing, but resets the line instruction from portrait values to landscape values to take advantage of the increased paper width.

Printing Problems:

1. Loosing a character at the end of a line.

Decrease the Page Width Land or Page Width Port value in the configuration dialog for your printer and press the Re-Config button to set that value.

2. Occasionally a page is printed containing only a line or two of code and without the standard header.

Decrease the Max. Lines Port or Max. Lines Land value in the configuration dialog for your printer and press the Re-Config button to set that value.

Landscape Printing Note:

Landscape printing requires much more memory on disk for temp files than portrait mode. Code in the form of standard txt files printed in portrait mode will generate average temp file sizes of about 1k per page. For an average project with about 80 to 100 pages of code you might expect your temp files to average about 100k for this project printed in portrait mode. Printed in landscape mode this same project with 80 to 100 pages of code will generate about 8 mgs of temp files to rotate all that text. These temp files are generated in your temp directory not in your permanent swap file!! Make sure you have enough memory on disk to print large projects in landscape mode. If you don't have enough memory for your temp files windows will not handle the shortage. Generally your system will lock up, and you will have to reboot and loose all your un-saved work. For all those people out there that have ram drives of one or two mgs for their temp files, your asking for major problems. For printing large projects in landscape, 5 to 6

mgs of free disk space is bare minimum. The more you have the faster and easier your printing jobs will proceed with all your Windows programs.

Some Dot matrix printers also accept the HP codes. If you feel the need to print sideways on your dot matrix use the Laser/Ink Jet designation and experiment away.

Launch at Startup

Launch at Startup: / Launch State:

ToolBox Pro will launch one program upon start up in addition to VB. If you have a program that you use in conjunction with Visual Basic and you wish to launch everytime Visual Basic is run, enter the full path and exe location in text box #3 labeled "Launch at Startup" and the start up state in #4 labeled "Launch State". Press F10 to browse directories and make your selection. For startup state: 1 = normal, 2 = minimize.

Button Launch Configuration

Button Launch: / Btn Launch State:

The left arrow button in ToolBox Pro is user configurable. In effect this is a quick launch button. If you have a program you use constantly within VB enter its full path in this text box and the launch state in the Btn Launch State text box. [1 = normal, 2 = minimized]. This function is identical to the launch programs list box. It uses the shell command, therefore anything including switches can be passed. For a program you use often this button is a convenience to save keystrokes.

Example adding the following entry { Notepad C:\VB\Constant.txt } the left launch button will display the text file "Constant.txt" in the Visual basic directory using notepad everytime you press this button. Or you can add the entry {C:\Windows\Winfile.exe} and have instant access to the file manager.

Loading your current project on start up

Current Project:

ToolBox Pro can launch your current project on start up. In the configuration dialog enter the path and *.mak file location in the appropriate text box. Pressing F10 will call up a Browse dialog box making configuration easier and error free. Once you find your desired file a double click will enter your choice in the appropriate text box. After you have completed the configuration process press the **RE-Config** button to write all changes to the configuration file.

Configuring ToolBar Buttons #1 & #2

ToolBar Btn #1; Btn #1 Cmd$ / Toolbar btn #2; Btn #2 Cmd$

The left two buttons on the floating ToolBar are user configurable; You can set them to use the shell function to launch a program, or the sendkeys function to access menu items. In the text boxes labeled ToolBar Btn #1 and ToolBar Btn #2 enter **SendKeys** to enable the SendKeys function for that button, or **Shell1** to enable the shell function in a normal state, or **Shell2** to enable the shell function in a minimized state.

In the Btn #1 Cmd$ and Btn #2 Cmd$ text boxes enter the command string. If your using the shell function there are examples in the VB help file. If you are using the SendKeys function to access a menu item **omit the quotation marks and enter only the keys to be sent.**

example File open would be Alt + File + Open; enter %FO in the text box.

If you select the shell function for these buttons the following examples will give you some guidelines :

Cmd$	Action
Notepad.exe	Launch Notepad
Notepad.exe c:\vb\constant.txt	Launch the text file constant.txt using notepad
c:\dos\comp.bat	Launches a batch file

Printer settings

Page Width Land; Page Width Port; Max. Lines Port; Max. Lines Land
These four text boxes allow you to customize the settings for your printer. The default settings should work for most printers. If you are loosing a character or two at the right of the page, or if the last two lines of a page are printed to a new page just adjust the appropriate value. Conversely if you are using non standard paper, your printer uses a smaller default type face, etc. and you have extra room on the page feel free to experiment to maximize your output per page.

Page Width Land;
This is the # of characters per page your printer is capable of printing in landscape mode.
default = 102

Max. Lines Land;
This is the # of lines per page your printer is capable of printing in landscape mode
default = 43

Page Width Port;
This is the # of characters per page in portrait mode.
default = 77

Max. Lines Port;
This is the # of lines per page your printer is capable of printing in portrait mode
default = 60

Note for Deskjet 500c and 550c color printer users:
This is specifically for 550c owners. I'm assuming that 500c printers have similar characteristics. If you have this printer you already know that to accommodate the color cartridge the page has been shifted up 1/4" and the default bottom margin has been increased to 7/8" . The values for this printer are
Page Width Land = 100
Page Width Port = 77
Max Lines Land = 43
Max Lines Port = 55

Minimize on start up and On Top Options

Minimize on start / Always on top:

These two check boxes control the properties of ToolBox Pro on start up. Minimize on start will start ToolBox Pro in a minimized state. Always on top will keep the ToolBox the top window at all times for easy access of its functions. If you minimize the ToolBox this property is disabled for the icon.

Choosing your ToolBar location

Tbox Loc:
This selection controls the position of the floating toolbar and ties it to the Visual Basic Main window depending on your monitor size and video card resolution. Possible values are 0 through 4.

640 x 480 using a 14 or 15 in monitor:
default = 4 0,1, 2 or 3 also available
800 x 600 using a 15 or 17 in monitor
default = 1
This places the ToolBar next to the Visual Basic ToolBar. In 640 resolution this covers the Twips position values at the right. If you need to use this for positioning you can minimize the floating ToolBar.

"0" eliminates the ToolBar
"2" places the floating ToolBar in the white menu section next to the help menu command
"3" places the floating ToolBar to the far right under the main window, above the project window.

1024 x 768 using a 17 in monitor;
default = 1 0, 1, 2, 3 or 4 available
This places the floating ToolBar next to Visual Basics default ToolBar.

"0" eliminates the ToolBar
"2" places the floating ToolBar in the white menu section next to the help menu command
"3" places the floating ToolBar to the far right under the main window, above the project window.

Feel free to experiment with placement for your way of working.

The best feature of the VB floating ToolBar is its floating property. It always stays on top no matter what window is active. Especially useful when developing on a 14 or 15 in monitor, you can now maximize your code windows and use the floating ToolBar to access menu functions for copy, cut , paste or any other menu item you wish to program in. Give it a try you'll never be able to go back.

How to configure Auto Save interval

AUTO SAVE:
The auto save feature of ToolBox Pro will save your project at specified intervals. **This function works only in design mode.** If your a debugging die hard that spend hours toggling between break and run, use the Save & Run button or press Ctrl+F5 to save your project before entering run mode.

Valid entries for this text box are: 100 to 9999 seconds.

Default = 1000 {Save at about 10 min. intervals}
0 [zero] Disables Auto Save

Configuring the launch listbox

Launch Programs:

ToolBox Pro allows you to launch any program from within Visual Basic. This feature allows you to stay within the programming environment and still have easy access to Icons, paint programs, text editors or any other program you may find necessary for developing your programs.

If this is the first time you are configuring this program, choose the "Configure Launch Program List" option and press the "Auto Setup" button.

This will search through your windows directory and your visual basic directory looking for standard programs such as IconWorks, Paint etc, and creates a standard setup for your system. Repeat this for the "Configure Help File" option button and you will have a basic configuration for your system.

Don't forget to press the Re-Config button before you exit to write all new information to the program configuration file.

Once you have a basic configuration in place you can use the browse dialog box to search for and add any program you desire to the list, or enter them manually. Add the title and location of programs you wish to access under each heading. Feel free to add as many as you wish, no limit. If you wish to edit or delete existing entries, double click to make your selection. Once selected you can make changes to the existing values and press the add button, or press the delete button to delete that entry. If you wish to re-order the list box, while your in the configuration dialog double click on a selection in the list. This will place that selection in the editing box. If you press "Add" without modification, the selection will be re-entered at the bottom of the list.

"Default Sample"

Title for List box	Exe with complete path
Icon Works	C:\Windows\Iconwrks.exe
Windows Paint	C:\Windows\Pbrush.exe
VB Tips	C:\vb\textfile\VB-Tips.exe
Call Help App	C:\vb\hc\Callhelp.exe
Control hWnd	C:\vb\vbfindid.exe
Notepad	Notepad.exe

Configuring the Help Listbox

Help Files

If you have any add-ons for Visual Basic one problem is sorting out all the accompanying help files. Generally you must assign them an icon and return to the program manager to access them. ToolBox Pro allows you to access these files from within VB. Select the "Configure Help" option button and call up the help file list box, enter the title and location of your help files.

If this is your first time configuring this program press the "Auto Setup" button. This will search your VB and your Windows\System directories for help files relating to Visual Basic or its third party add-ons and create a basic configuration for your system and programs. Once this basic system configuration is complete you can add or delete any entry to further customize the program for your working environment.

Title for list box = Title of help file to be entered into the ToolBox Pro selection list box.
Help File Location = Path and name of Help file.

This function will only launch Windows help files using the Windows help engine, don't try to add programs to this list.

"Default Sample"

Title for list box	Help File Location
VB Knowledge	C:\vb\vbknowlg.hlp
Control XRef	C:\vb\ctrlref.hlp
Setup Kit	C:\vb\setupkit\setupkit.hlp
Widgets #2	SS3d2.hlp
Widgets #3,	SS3d3.hlp
Win Api	C:\vb\winapi\winsdk.hlp
API XRef	C:\vb\winapi\apixref.hlp

Important:

After you finish making changes to the ToolBox Pro Configuration file press the Re-Config button to write all new information to the program configuration file and change all global variables.

Identifying and correcting Configuration Errors

Correcting Errors:

If you utilize the Auto Setup and the Browse [F10] functions within the configuration dialog to enter your program choices and their paths, errors should be held to a minimum. For all other options as you tab or click from one text box to another, basic instructions and defaults are displayed. If something doesn't run properly check first for spelling or extra spaces.
Next check to see if you have deleted or moved the file ToolBox Pro is trying to access.

In the Help File Function if you get the big blue error message check your entry for proper spelling and the correct path. This is the standard error displayed by windows if the winhelp function can't find your help file.

If you have any comments or suggestions for future versions you can reach me on COMPUSERVE, Or by writing to me at the address below.

COMPUSERVE 71042,1566

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

Ordering Info

Thanks for trying Visual Basic ToolBox Pro Version 3.0

Only $49.95 & shipping and handling

US and Canadian Orders		**Foreign Orders**	
Reg.	$49.95	Reg.	$49.95
S&H	$ 5.00	S&H	$ 7.00
Total	$54.95	Total	$56.95

If you are paying by check or money order
Make checks payable to:
Hal Jurcik
7819 Vanport Ave.
Whittier, CA. 90606
COMPUSERVE ID: 71042,1566

If you wish to Order using Master card, Visa, or American Express you may call our registration service:

Ask for Product Item # 10891

Public Software Library Registration Service:
800-242-4775

For foreign orders
713-524-6394
Fax# 713-524-6398

If you are a member of Compuserve you can also place your order via Cis E-Mail
Send your Name, Complete mailing address and Credit Card information to:
PSL Registration Service
CIS ID # 71355,470

Note: These phone numbers are for CREDIT CARD ORDERS ONLY !!
The operators are order takers only. They cannot give any specific answers or support for any program. Any questions about this program, volume discounts, dealer pricing, technical questions or problems of any kind Contact me on COMPUSERVE or at the above address.

Upgrades for Registered users of Version 2.0 of ToolBox Pro.

If you are a registered user of VBToolBox Pro 2.0 you can upgrade to the newest version for $29.95. $24.95 + $5.00 shipping and handling. All upgrades are handled **through my office ONLY**. The registration service I use is not setup for upgrade sales.

Make checks payable to:
Hal Jurcik
7819 Vanport Ave.
Whittier, CA. 90606

Features:

1. Code Manager, catalogs all your project code
2. API Manager, catalogs all Windows and add-on API Decl, DLL Decl, Type Decl, and Global constants
3. User Configurable PopUp Code Menu
4. User Configurable PopUp Menu
5. User Configurable Floating ToolBar
6. ToolBox Pro Instant Watch
7. Save & Run; Saves you work before entering run mode
8. Auto Save, Saves your work at specified intervals
9. Load your current project on start up
10. Automatically start 2nd program on start up
11. Printing capabilities for everything from the smallest code fragment to your entire project.
12. Add VBx files to your project with the click of a mouse
13. List of all active windows tasks and their corresponding Task ID #;s and hWnd's
14. VB Spy program to find the hWnd's of any control, form, or window along with its parent /owner
15. Launch any program from within Visual Basic
16. Easy access to all third party help files from within VB
17. ToolBox Pro Help file
18. Ability to automatically save your project as a series of text files

---

Printing Code

The Visual Basic ToolBox has two separate printing functions:

Print a code fragment:

Even with the best monitors, reading code is difficult on screen. The Print clip function allows you to select text with the mouse and print only that fragment for easier analysis. Two methods are available for accessing this function. After selecting a block of text with the mouse press the "F6" function key. This hot key will call the active form you are working on and print the text you have highlighted. A button for this function is also available in the "Text Operations" dialog box.

All code formatting is retained. The form name, date, time and page number is included in the header for future reference. Text is wrapped automatically. Of course the printer must be ready for this function to work.

Printing text files:

The text operations button in ToolBox Pro has three functions. The printing capabilities are contained in the Create Text File operation. As each form is saved as a text file, an entry is made in a list box to the left of the ToolBox Pro. Once all files have been saved as text you may select any or all the files in the list box for printing. Click the list box with the right mouse button and choose Portrait or Landscape orientation to print the selected files.

Caution:

Creating a new set of text files will overwrite all older versions in that directory. If you wish to save an older version of your text files, either move them to another directory or floppy disk before creating the new set, or create the new set in a different directory.

Naming convention for "frm" and "bas" files:

DO NOT give the same name to a "bas" file and a "frm". Saving your project as text files assigns the standard "txt" extension to all files, therefore the second file will overwrite the first!!! If you have two files with identical names an error message will appear asking you to rename the second file and re run the Save Text operation.

Notes on using landscape mode:

Landscape printing requires much more memory on disk for temp files than portrait mode. Code in the form of standard txt files printed in portrait mode will generate average temp file sizes of about 1k per page. For an average project with about 80 to 100 pages of code you might expect your temp files to average about 100k for this project printed in portrait mode. Printed in landscape mode this same project with 80 to 100 pages of code will generate about 8 mgs of temp files to rotate all that text. These temp files are generated in your temp directory

not in your permanent swap file!! Make sure you have enough memory on disk to print large projects in landscape mode. If you don't have enough memory for your temp files, windows will not handle the shortage. Generally your system will lock up, and you will have to reboot and loose all your un-saved work. For all those people out there that have ram drives of one or two mgs for their temp files, your asking for major problems. For printing large projects in landscape, 5 to 6 mgs of free disk space is bare minimum. The more you have the faster and easier your printing jobs will proceed with all your Windows programs.

For information on configuring ToolBox Pro for your printer type also see:
Configuring ToolBox Pro

Saving Code as text files

The Save code as text file function gives you the ability to automatically save your project as a series of text files in one operation to any directory you choose. When you click on the Create/Print Text File button a dialog box appears to allow selection of a directory. There is no default, you must explicitly choose a directory. When creating text files all previous versions in that directory will be overwritten, if you wish to save and compare different versions make sure to save to an alternate directory!! As each file is created its name is added to a list box. Once the operation is complete you may use this list to print one or more files of your project. Make your choices using the left mouse button, once your happy with your selection click on the list with your right mouse button to send those files to the printer. If you don't wish to print any project files at this time, a right click on the list with no selections will close the list.

Usage:

a) Text files make an easy backup of your projects, if your making major changes to a program you can use these files as a reference, or restore your project to its original version if desired.

b) Automatically indexes the project in the code manager for easy access to reusable code

c) Copy text files to floppy for circulation among extended work groups.

d) Even on the best monitors viewing code on screen is difficult. You can use ToolBox Pro's printing functions to make hard copies of your project code for easier examination and circulation.

Also see Printing Code for a more detailed explanation

Printing Code

Program Hotkeys

I've included two Hotkeys for the most used functions of ToolBox Pro to increase usability and convenience.

Ctrl+F5:

This is the hotkey for the Save & Run procedure. It can be used in place of F5 or Shift+F5. This hotkey will save all project files before entering run mode protecting you from lost work and frustration if the system goes down. Both F5 and Shift+F5 remain functional therefore if you wish to test a piece of code without saving, both options are available to you.

F6:

This hotkey automatically prints any text you have selected within a form. Great for examining code while debugging. Select the text you wish to print with the mouse, and press F6, **It's that simple.**

Ctrl+F9:

This hotkey accepts the defaults for instant watch and sets your watch variables with one action.

Installation

DISCLAIMER:

ToolBox Pro 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 ToolBox Pro may be put, no warranty of fitness for a particular purpose is offered.

Installation:

File List:

Program files,

VB3_TBOX.HLP	This File
VB3_TBOX.DOC	ToolBox Manual {Winword Format}
VB3_TBOX.EXE	ToolBox Pro executable
TOOLBOX3.CFG	Configuration file
TBOX1.$$$	Configuration file
TBOX2.$$$	Configuration file
THREED.VBX	System file
SS3D2.VBX	System file
SPIN.VBX	System file
HOTKEY.DLL	System file
PPORIENT.DLL	System file
TSBCPRO.VBX	System file
SPYDLL.DLL	System file
API_DECL.MGR	Code Manager File
API_TYPE.MGR	Code Manager File
WIN_GLOB.MGR	Code Manager File
DATAGLOB.MGR	Code Manager File
VB_GLOB.MGR	Code Manager File
SORTS.LIB	Code Manager Sample Lib
SYS.LIB	Code Manager Sample Lib
PRINTER.LIB	Code Manager Sample Lib
INI_MOD1.LIB	Code Manager Sample Lib
FILE.LIB	Code Manager Sample Lib

Installation program files

README.TXT	Readme 1st file

SETUP.EXE	Setup initialization
SETUP.LST	Setup initialization file
SETUPKIT.DLL	System file
VBSETUP.EXE	ToolBox Pro Setup program

To install ToolBox Pro 3.0 run the setup program from the program manager or file manager and follow the Online instructions. Installation including icon creation is automatic.

The installation program is set up to work properly with Windows Program Manager and most other desktop shells. Some desktop shells do not process the DDE close message properly. If you receive a DDE error message a few minutes after installation choose "Close". The program is installed properly, for some reason your shell did not close the DDE connection after installation.

Configuring ToolBox Pro for your system

 ToolBar

The new ToolBox Pro ToolBar adds 8 buttons to Visual Basics existing one. The purpose of this extra ToolBar is to add those functions that MS neglected in the original. The left two buttons are user configurable for either Sendkeys or Shell functions. The most important feature of this ToolBar is that it will always float on top of code windows. Now you can edit in a maximized code window and have access to your editing functions. ToolBox Pro will monitor your programming environment to integrate closely with VB 2.0. The ToolBar will recede in run mode, when a help file or program unrelated to VB is active and minimize or exit in conjunction with Visual Basic.

Note:

Cut, Copy, Paste and instant watch functions etc.. in the floating ToolBar function only in design and or break modes, just like their menu counterparts. I cannot foresee what functions you will program into the user configurable buttons, therefore I cannot disable a button if that function is unavailable in one of VB's three modes. If you press a button and nothing happens, ...

For information on ToolBar configuration and positioning also see:

Configuration

 Btn #1:

User configurable, This button can be configured to sendkeys to activate menu functions or for the shell function to launch any program.

 Btn #2:

User configurable, This button can be configured to sendkeys to activate menu functions or for the shell function to launch any program.

Configuring ToolBar Buttons #1 & #2

 Btn #3:

"Cut" Deletes any selected text and copies to the clipboard.

 Btn #4

Copies any selected text to the clipboard.

 Btn #5:

Save and Run; This button will save your project and then place VB in run mode. Use this in place of the run command and you will never again loose code if the system locks up or GPF's. For added convenience I have added a global hot key to this feature [Ctrl + F5]. You now have three hot keys for running your programs, F5 for Run; Shift F5 to continue from break mode and Ctrl + F5 for Save Project and Run.

 Btn #6:
Auto Watch: The instant watch button MS provided is anything but instant. It calls up two confirmation dialog boxes. If you use the defaults this button will set your watch variables in one click. For keyboard users this function also has a global hotkey. [Ctrl+F9].

MS instant watch = Shift+F9
ToolBox Pro instant watch = Ctrl+F9

 Btn #7
Print selected text. In design or break mode use the mouse to select any portion of your code and this function will copy it to the clipboard and print. [Global hotkey = F6]
Printing Code

 Btn #8:
Loads the ToolBox Code Manager
Code Manager

 Btn #9:
Minimizes the ToolBar and turns off floating property.

ToolBar Min button

Minimizes the ToolBar and turns off floating property.

Load Code Manager

Loads the ToolBox Code Manager
Code Manager

Print Clip

This button will print the any text you have selected in the current code window. In design or break mode use the mouse to select any portion of your code and this function will send that segment to the printer along with the date, time, and code window title for future reference.

[Global hotkey = F6]
Printing Code

ToolBar Buttons #1 & #2

The left two buttons on this ToolBar are user configurable, These buttons can be configured to sendkeys to activate menu functions or for the shell function to launch any program.
Configuring ToolBar Buttons #1 & #2

Save and Run Button

This button will save your project and then place VB in run mode. Use this in place of the run command and you will never again loose code if the system locks up or GPF's. For added convenience I have added a global hot key to this feature [Ctrl + F5]. You now have three hot keys for running your programs, F5 for Run; Shift F5 to continue from break mode and Ctrl + F5 for Save Project and Run.

Cut

"Cut" Deletes any selected text and copies to the clipboard.

Copy

Copies any selected text to the clipboard.

ToolBox Pro Instant Watch

This button accepts the default local settings
and instantly adds the selected variable to
the debug window saving you keystrokes.
For keyboard users this function also
has a global hotkey. [Ctrl+F9].

MS instant watch = Shift+F9
ToolBox Pro instant watch = Ctrl+F9