VB Add-On Witch 1.1

Introduction

VB Add-On Witch helps you bringing order into chaos. With VBXes broadly available even the average VB programmer uses several add-on libraries with usually more than just one VBX and related help file.

To use one of these libraries or browse its help file he/she has to load the VBX through a standard dialog box or switch to the Windows file manager. Both can be quit time consuming, since loading the Add-dialog box with the contents of the Windows system directory might take a while. Fortunately there is VB Add-On Witch. With it loading a VBX into your project or look up a related help file is a snap. The witch´s main features:

n Provide quick access to all VBX/DLL, help files and any application with a simple directory tree of your favorite add-ons.

n Load any VB project without reverting to browsing through DOS directories.

n Create VB project templates for various purposes, e.g. database programming, spreadsheet applications etc. When VB Add-On Witch is loaded, every time you use the VB ´New Project´ menu command, you can choose among your templates to start your new project with.

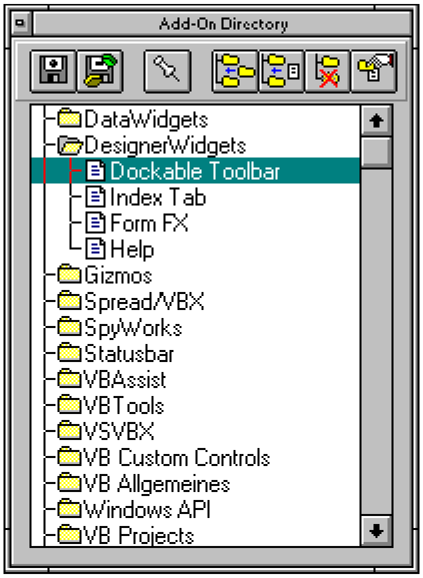
What´s New in Version 1.1

- Functions to build and edit the Add-On tree were added to the the Add-On directory window.
- Files to be added from the Add-On tree don´t necessarily have to be listed with a complete path anymore. VB Add-On Witch tries to locate them in the Windows/System directory or in any PATH-directory.
- The VB Add-On Witch main window can now be moved to any place on screen and will be positioned there automatically the next time you run the program.

See for details:

Add-On Directory Window
Add-On Directory Definition

Add-On Directory Window



The add-on directory displays a hierarchical view of all the add-on products, help files, VB project files and executables.
The ´directory tree´ does not mirror your file directory but is a depiction of the structure you set up in the INI-file.
There are several advantages you gain by accessing your tools through the add-on directory:

1. Regardless of how much all your VBX, .HLP, .BAS etc. files are scattered on your harddisk, by setting up the add-on tree in the INI-file you determine exactly how many mouse clicks they are away when you need them in your VB project.

2. By double clicking just once on a tree item you can add multiple files to your VB project, e.g. a VBX and its related .BAS file.

3. Help files, example .MAK files and any executables are readily accessible.

Lets you choose a file to add to your VB project using the standard file dialog box. Use this if you have not yet placed a particular file in the add-on tree.

Removes mutliple files from your VB project.
If you have selected an entry in the add-on directory, all the files it stands for (i.e. that would be added if you doubleclicked on it) will be removed from the VB project.
In case a VBX cannot be removed due to still being used on some form, all removals completed so far will be undone.

Keeps the add-on directory on the desktop.
By pressing this button the add-on directory will not be closed after having added or removed files.

Insert an add-on directory at the current location in the tree.
The new directory will usually be added above the current item and on the same level - except when the current item itself is a directory and is open: then the new directory will be added immediately below it and indented.

Insert a file or a list of files at the current location in the tree.
The new item will usually be added above the current one and on the same level - except when the current item is a directory and is open: then the new item will be added immediately below it and indented.

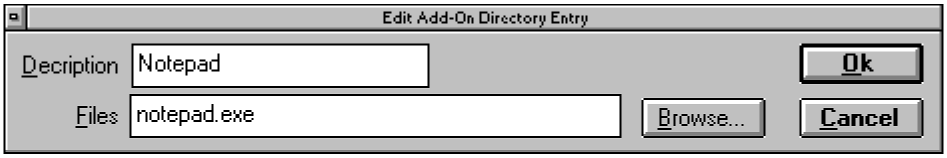
Delete the current item of the tree. If the item is a directory VB Add-On Witch will ask for confirmation if the directory is not empty.

Edit the current item

For details on the definition of the add-on hierarchy see:

Add-On Directory Definition

Editing an Add-On Directory Item



Decription: Notepad

Files: notepad.exe

Description

Description of the entry to be shown in the Add-On directory tree.

Files

When editing a file entry only: Either a list of files to be included in a VB project when double-clicking on this entry, or the command line for an executable file, or a help file. See examples below:

```
spread.vbx | c:\spread\vb_const.txt
```

A VBX and an auxillary file to be included into a VB project. The VBX is listed without a path so VB Add-On Witch will try to locate it in the Windows/System directory or in any of the PATH-directories. Files must be separated with a ´|´.

```
notepad.exe \autoexec.bat
```

Command line for an executable file. The first file must have a .PIF, .BAT, .COM or .EXE extension. The program will be run when doubleclicking on the item.

```
\vb\vbapi.hlp
```

The helpfile will be shown using WINHELP.EXE when doubleclicking on the item.

```
\myprog\myprog.mak
```

VB Add-On Witch will open the VB project when doubleclicking on the item.

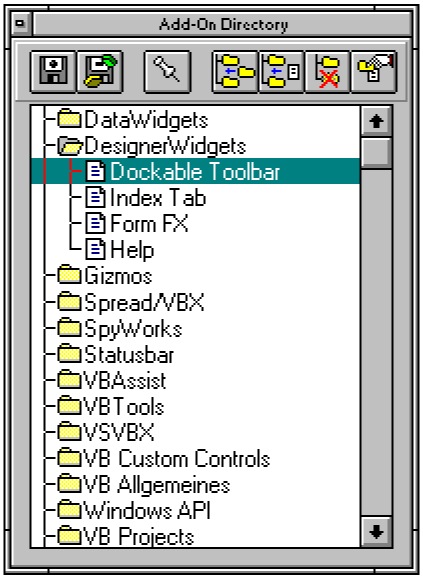
Browse...

When editing a file entry only: Opens a standard file dialog box and lets you select a file to be appended to the current list of files. Unless the file is located in the Windows/System directory or in any PATH-directory the full path will be included in the filename. A ´|´ is automatically put in front of the filename to separate it from the other filenames already present.

Note:
While the edit window is open the VB main window is disabled to prevent you from closing or minimizing it while this MODAL window is open.

Add-On Directory Definition

The directory definition for all add-on tools is located in the AOWITCH.INI file. How you set up the directory is important for the speed with which you will be able to add your favorite VBX to VB projects or are able to look at a related help file.



For parts of the add-on directory tree shown above the following is the definition you set up in the AOWITCH.INI file:

```
[Add-On Directory]
1=0;DataWidgets
2=1;Datagrid & Combobox;q:\bin\ssdatwdg\ssdata.bas | ssdata2.vbx
3=1;Enhanced Data Ctrl & Cmd Button;ssdata1.vbx | q:
\bin\ssdatwdg\ssdata.bas
4=1;Option Button;q:\bin\windows\ssdata3.vbx | q:\bin\ssdatwdg\ssdata.bas
5=1;Help;q:\bin\ssdatwdg\ssdatwdg.hlp
...
10=0;DesignerWidgets
30=0;SpyWorks
...
40=0;VBAssist.VBX;vbassist.vbx
...
100=0;VB Projects
101=1;VB Trans;q:\daten\vb\vbtrans\vbtrans.mak

110=0;Notepad;notepad.exe
```

The syntax for an entry:

```
<item number> "=" <level> ";" <description> [ ";" <file list> ]

<item number>
```

A unique number identifying the add-on directory entry for the GetPrivateProfileString() API function.

`<level>`

States where in the directory tree the item is located. The ´Data Widgets´ item is on level 0, the items subsumed under it are on level 1 and immediately follow it. Whenever an directory definition item has entries directly listed after it with a higher level it is assumed to be a ´directory´ (i.e. not a file to be included in a project) and will be depicted using a small folder.
There is no limit to the number of tree levels you can nest other than that set by the MSOUTLINE.VBX. Just be careful not to skip a level when defining the hierarchy.

`<description>`

A string describing the entry; it will be displayed in the outline window as shown above.

`<file list>`

An optional list of files to be included into a VB project (see for example the ´Datagrid´ item in the above example). The files are separated by ´|´ characters. Thus by selecting just one item in the add-on directory you can add multiple files to you VB project.

- If the first file list entry is a .HLP file WINHELP.EXE will be called to display the file.
- If the first entry has a .EXE entension it will be executed (see ´Notepad´ item above). You can even pass command line parameters to it by writing them right after the .EXE filename, e.g.

  ```
  110=0;Notepad;notepad.exe readme.txt
  ```
- If the first entry is a .MAK file, the current VB project will be closed and the file listed in the entry opened. This is similar to using the ´Open Project´ VB menu command.
- If a file is listed without a path VB Add-On Witch tries to locate it either in the Windows or System directory or in one of the PATH-directories. It will then be included in the VB project with its full path prepended.

See also:

Editing an Add-On Directory Item

Installation

1. Copy all files you received into the VB Add-On Witch application directory. DLL/VBX files may be placed into the Windows System directory.
2. AOWITCH.INI can either remain in the application directory or may be moved to the Windows directory.
3. Adapt AOWITCH.INI to your needs. At least review the entries in the [System] section.
4. Create Program Manager application entries using a command line as shown below. When run VB Add-On Witch will first start VB.EXE (or whatever application is set up under Startup= in AOWITCH.INI) and pass any command line parameters to it.

```
\aowitch\aowitch.exe myproj.mak
```

Requirements

VB Add-On Witch - when received as a single ZIP file - requires Visual Basic 3.0 Professional to be installed on your system, i.e. VBRUN300.DLL, MSOUTLINE.VBX etc.

See also:

AOWITCH.INI Description
Add-On Directory Definition

INI-File

The VB Add-On Witch INI-file consists of several sections that will be described in turn. It is called AOWITCH.INI and can be placed in the VB Add-On Witch application directory or in the Windows system directory.

[System]

```
Startup=q:\bin\vb\vb.exe
VBMenuAdd=%fd
VBMenuRemove=%fr
VBMenuOpen=%fo
OpenProjectDialogTitle=Open Project
AutoloadProjectSignature=Project1;Form1.frm;vbalink.dll
Help=q:\bin\windows
```

Startup=

Defines which program to run when started. Since VB Add-On Witch requires Visual Basic to be running you should list either VB.EXE or any other program resulting in VB running (e.g. VBASSIST.EXE from Sheridan Software).

VBMenuAdd=

Keystrokes to access the ´Add file...´ VB menu command. Use VB´s SendKeys-command keystroke syntax. In the above example ´%f´ (Alt-F) stands for opening the ´File´ menu, ´d´ is for actually selecting the ´Add File...´ entry.

VBMenuRemove=

Keystrokes to access the ´Remove File´ VB menu command. Use VB´s SendKeys-command keystroke syntax.

VBMenuOpen=

Keystrokes to access the ´Open Project...´ VB menu command. Use VB´s SendKeys-command keystroke syntax.

OpenProjectDialogTitle=

List here the caption text of the dialog box displayed to let you select a VB project. This entry enables VB Add-On Witch to recognize this particular dialog box for switching to another project without requiring any user input.

(There does not seem to be any other way to distinguish the different dialog boxes used from one another, at least their class names are the same).

AutoloadProjectSignature=

In order to recognize if the ´New Project´ command has been aborted by the user, VB Add-On Witch scans the VB project window before and after command execution. By comparing ´project signatures ´ it tries to determine if the project has changed (i.e. the command had not been aborted) or stayed the same. The ´project signature´ is a list of filenames starting with the project name, e.g. ´MyProject;MyForm1.frm;MyModul.bas;Threed.vbx´.
To enable VB Add-On Witch to recognize a freshly created project you have to provide its ´project signature´ with the above INI-file entry. Be sure to:

1 Separate all names in the ´signature´ with ´;´

2 Use no blanks in the ´project signature´.

3 List the entries in the autoload.mak file in their order of appearance. Strip off any path information. Use notepad.exe to view the autoload.mak.

4 Be sure to include ´Project1;Form1.frm´ as the first two entries. (Beware of national differences!)

The above example includes vbalink.dll which is only necessary for VBAssist from Sheridan Software to work properly.

Help=

Path of help file AOWITCH.HLP. Usually this entry is not used at all, since the help file will be expected and installed in the app.path.
When used, please don´t write a ´\´ at the end of the path.

[Defaults]

```
optCloseAfterAdd=0
AddFileFilter=Basic Files (*.bas;*.frm)|*.bas; *.frm|Libraries
(*.vbx;*.dll)|*.vbx; *.dll|Text Modules (*.txt)|*.txt|All (*.*)|*.*
frmLib=5955;3195;2355;3990
position=1
useVBProjectNew=0
```

optCloseAfterAdd=

Determines if the add-on directory window will be closed immediately after adding/removing files to/from the VB project. Possible values:
0   : keep window on desktop
-1 : close window after add/remove

See also:

Add-On Directory

AddFileFilter=

Lets you set up your preferences for the file filter used in the dialog box for adding a file not listed in the add-on directory. Use the syntax required for the ´Filter´ property of the cmddialog.vbx. The default will be the first entry.

frmLib=

*** This entry is for internal use and alteration only! ***
It saves the last position of the add-on directory window.

position=

Determines if VB Add-On Witch is shown on the right or left side of the VB main window. VB Add-On Witch automatically attaches itself to the VB main window when started. Possible values:
0 : left side
1 : right side

useVBProjectNew=

Determines if project templates will be offered after a ´New Project´ command. Possible values:
0 : no use of project templates
-1 : project templates will be offered

See also:

Project Templates

[Add-On Directory]

Add-On Directory Definition

[Project Templates]

```
Normal=q:\bin\windows\threed.vbx | q:\daten\vb\utils\util\const.bas
Database=q:\bin\windows\ssdata2.vbx | q:\bin\windows\ssdata1.vbx | q:
\bin\ssdatwdg\ssdata.bas | q:\daten\vb\utils\util\const.bas
MDI Application=q:\bin\windows\ssdocktb.vbx | q:
\daten\vb\utils\util\const.bas
Spyworks=q:\bin\windows\sbc.vbx | q:\bin\windows\cbk.vbx | q:
\bin\spyworks\spywin.bas | q:\daten\vb\utils\util\const.bas
```

Syntax

```
<description> "=" <filename> { "|" <filename> }
```

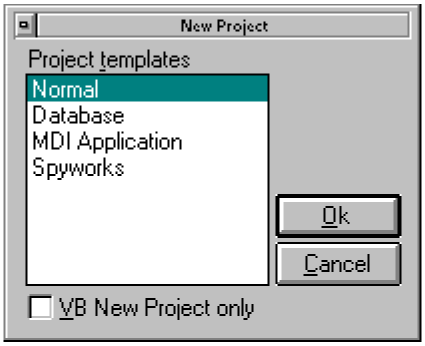
`<description>`

A string describing the template that will be displayed in the project template window presented during the ´New Project´ VB menu command.

`<filename>`

Name of a file to be included in the new VB project. All filenames should be fully qualified.

Project Templates



This windows is shown when you use the ´New Project´ menu command of the VB environment. It lets you choose from a list of project templates you set up in the INI-file.

Project templates are a means of getting a head start whenever you begin a new project. Similar to the VB autoload.mak project templates already include as many VBX, .BAS or .FRM files as you want. But VB Add-On Witch so to speak places an unlimited number of prefab projects at your fingertips. For example you can set up templates for database applications including some bound data grid VBX, related controls and a constant definition .BAS, or you could define a template for MDI programs including a dockable toolbar VBX and the OLE control. Get lazy, use templates.

The Steps to Follow

n Set up your templates in the INI-file

INI-File

n Clear your autoload.mak from all files using notepad.exe. Thus you gain complete control over what is loaded into a new project by just using templates.

n Call the VB ´New Project´ menu command with VB Add-On Witch running. VB will create an empty project with just a ´Form1´ entry. Then the above ´New Project´ window appears and you can select a template to fill the so far empty VB project.

How to Use It

`Ok`

Press the Ok button or doubleclick on a template to select it as the basis for you new VB project.

`Cancel`

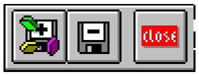
Closes the template window. The new VB project created remains in the current incomplete state, i.e. probably empty except for the ´Form1´ entry.

`VB New Project only`

Select this checkbox if you never want to use templates. The ´New Project´ menu command will not show the template window again.

To undo the effect of this checkbox accidentally selected please change the corresponding entry in the INI-file.

Main Window



VB Add-On Witch can only be run in conjunction with the Visual Basic development environment. Its functionallity is accessed by clicking on the buttons displayed.
VB Add-On Witch attaches itself to the VB main window.

Calls up the add-on directory window to let you quickly add files to your current VB project, display help files or run any application.

Add-On Directory Window

Removes the currently selected file from the VB project. It is similar to the ´Remove file´ menu entry of the VB environment.

Closes VB Add-On Witch and restores the original VB ´New Project´ menu item functionality.

For details on positioning the VB Add-On Witch window and calling VB on start up see the INI-file description:

INI-file

Command Line Parameters

```
-A
```

When started VB Add-On Witch will not attempt to run the startup application listed in the [System] section of AOWITCH.INI.

```
-H
```

VB Add-On Witch will not be hidden in VB´s Run or Break Mode.

```
myproj.mak
```

Any other command line parameters, e.g. a .MAK file name, will be passed to the startup application.

It is not allowed to mix the -A/-H switches with any other command line parameters.

Registration

You cannot yet formally register for VB Add-On Witch. The software is still in its beta stage.

If you are interested in registering, please send a personal mail to us. We´ll then put you on our list and you will be the first to hear about the final release.
If you have any suggestions or bugs you would like to report please do the same. Thanks a lot in advance for you contribution.

VB Add-On Witch is developed by:

Mey & Westphal
RIPOSTE Software GbR
Schottweg 1
22087 Hamburg
Germany

Tel +49 (0)40 - 22 17 23
Fax +49 (0)40 - 22 17 72
CIS 100111,305