

VBcheck v1.01

Copyright (c) 1994 Explosive Software

Simple, effective and inexpensive source code control for Microsoft's Visual Basic.

VBcheck will:

1. Allow you to manage a Visual Basic project's source code between a team of developers
2. Take only a few minutes to set-up and learn

What does VBcheck do?

VBcheck allows developers to select a form or module in the Visual Basic (VB) *Project Window* and at the press of a button:

- Check it out for modification
- Check it in following modification and testing
- Get the latest version of it

Your master VB source code is stored in read-only format. In the VB *Project Window* master files are displayed with a red padlock symbol whilst checked-out files without the red padlock are easily identified.

When a developer checks out a file, VBcheck checks to see if anyone else has that file checked out first. If so it informs you who has checked it out. If not, a copy of the read-only master code is created and replaces the master version in your VB project. Other developers will still be using the master version.

When a developer checks a file in, a backup of the master file is created and the master file is replaced with the version being checked-in which is made read-only.

When VBcheck is closed it displays a list of all the files which you still have checked-out.

VBcheck works with files saved in Text or Binary format.

Backup files have the following extensions:

.BAK	for	.FRM	
.BAX	for	.FRX	(if the Form was saved in text format)
.BAC	for	.BAS	

How to use VBcheck

Visual Basic must be in design mode (VB titlebar text ends with [design], i.e. not running) before using any of VBcheck's options, otherwise an error occurs "Cannot find Visual Basic in design mode".

To check a file out:

- select the file in the Visual Basic *Project Window*
- click on the **Out** button

To check a file in:

- make sure that the project works correctly
- select the file in the Visual Basic *Project Window*
- click on the **In** button

Note: also check in other modified files which depend on it otherwise other developers who use the new version will not be able to run the project.

To get the latest version of a file (after someone else has checked it in):

- select the file in the Visual Basic *Project Window*
- click on the **New** button

Note that the **Out**, **In** and **New** buttons all save the Visual Basic project before performing their intended function. Therefore unsaved changes will be saved and if a read-only file has been changed then you will be notified and will have to take appropriate action before retrying.

To discard changes after checking a file out:

- select the file in the Visual Basic *Project Window*
- remove the file from the VB project from VB's File menu without saving it
- add the master file into the VB project from VB's File menu

To discard changes made without having checked the file out:

- select the file in the Visual Basic *Project Window*
- remove the file from the VB project from VB's File menu without saving it
- add the master file back into the VB project from VB's File menu

To add a new file to the project:

- create the new form or module as normal
- save the new file into your own directory (see set-up section below)
- check file in when fully working

To find out what files you have checked out:

- try to close VBcheck
- a list of all the files which you have checked-out is displayed
- select No to the message "Do you want to close?" to return to VBcheck

How to set-up VBcheck

Create a directory in a shared file area on your file server for your VB project's source code
e.g. P:\SOURCE\EXAMPLE

Place any existing source code in this directory and make it read-only to protect it (read-only files are displayed within the VB *Project Window* with a red padlock symbol which make master source code easily distinguishable from file that you have checked-out).

Create a subdirectory for each of the developers
e.g. P:\SOURCE\EXAMPLE\JANET

Checked-out source code will be stored in these subdirectories. The name of the subdirectory will

be displayed in a message if someone else has checked out a file that you are trying to check out
e.g. "Janet has this file checked out".

Set-up a project file on each developers PC
e.g. C:\PROGRAMS\EXAMPLE\EXAMPLE.MAK

Create VBCHECK.INI (modify the example supplied according to the notes below) on each PC in the Windows directory. The 4 developer and unrestricted versions have a built-in utility for setting up these details.

For the example directories used above, VBCHECK.INI would contain the following lines:

```
[VBcheck]
MyName=Janet
Current=Example
```

```
[Projects]
Example=C:\PROGRAMS\EXAMPLE\EXAMPLE.MAK P:\SOURCE\EXAMPLE
```

MyName must be unique within the team and be the same as one of the subdirectories in the master source code directory.

If VBCHECK.INI can't be found or has missing or incorrect settings then VBcheck will produce errors.

As a VB developer, it is assumed that you will have VBRUN300.DLL, THREED.VBX and PICCLIP.VBX as these may not be supplied with VBcheck.

How to purchase/upgrade VBcheck

Prices (per user):

	less than 10 copies	more than 10 copies
2 developer option	£10 or \$15	£10 or \$15
4 developer option	£15 or \$25	£10 or \$15
unrestricted option	£20 or \$30 (min 5 copies)	£15 or \$25
all upgrades	£10 or \$15	£5 or \$10

(no dollar cheques please - U.K. Sterling cheques, Eurocheques and cash only)

2 and 4 developer options restrict the number of developers that can work on a given VB project.

4 developer and unrestricted options also include:

- on-line help
- facility to switch between projects
- built-in configuration set-up
- bug fix and upgrade information

Please make cheques payable to:

J.E.Colton

please send to: 2 Lessingham Road
Widnes
Cheshire WA8 9FU
England