VERSIONS/VB Help

Glossary Technical Support

How to use VERSIONS/VB

Learning VERSIONS/VB

Using VERSIONS/VB

VERSIONS/VB Project Window

General Information

Reference

Version Control Environment

Common Questions and Answers

Messages

Copyright Notice

Viewing File History

Produce a history report for the selected file. A history report displays detailed information about all the file versions stored in the versions directory.

Dialog Box Options

Project
Displays the current project name and path.

Versions Directory
Project files contained in the versions directory appear in the list box. Select the file for which the report will be produced.

Report on Version Attributes
Select the attributes to be included in the report by activating the appropriate check box. By default, all attributes are included.

See Also
Project Reporting

Checking In Files

Check In a file to store a copy of the current working version in the versions directory.

Dialog Box Options

Project
Displays the current project name and path.

New/Existing
New -- Displays only new project files (files that have never been checked in before).

Existing -- Displays only existing project files (files that have previously been checked in).

Visual Basic Project Files & Selected Files
Project working files appear in the list box on the left. Files to be checked in appear in the list box on the right. VERSIONS/VB automatically enters those files that have changed into the Selected Files list box.

> Copies the highlighted file from the Project Files list box to the Selected Files list box.

>> Copies all files from the Project Files list box to the Selected Files list box.

<< Clears all files from the Selected Files list box.

< Clears the highlighted file from the Selected Files list box.

Permanent
When checked, designates all file versions being checked in as Permanent.

Keep Locked
When checked, retains your lock on the file so you can continue to work on it.

Duplicate OK
When checked, allows you to check in a file even if it is the same as the most recently checked in version in the versions directory.

Maximum # of Versions
Enter a new maximum number of versions for the files (only available when checking in new files).

Version Label
(Optional) Assign a name (up to 32 characters) to the current version. This allows you to retrieve the version by name rather than number.

Description/Reason for Checking In This Version
If you are checking in new files, enter a description (up to 256 characters) of the files. If you are checking in existing files, enter a reason for checking in this version (up to 128 characters). Descriptions and reasons are attached to all files being checked in during a single operation.

Suggest
VERSIONS/VB "suggests" which files to check in by entering changed files into the Selected Files list box automatically. If you have made changes and want to return to the VERSIONS/VB suggestions, click Suggest.

See Also
Check In Hints

Checking Out Files

Check out files to retrieve a file version from the versions directory.

Dialog Box Options

Project

Displays the current project name and path.

Files in Versions Directory & Selected Files

Project files contained in the versions directory appear in the list box on the left. Files to be checked out appear in the list box on the right. VERSIONS/VB automatically enters those files that have changed into the Selected Files list box.

> Copies the highlighted file from the Versions Directory list box to the Selected Files list box.

>> Copies all files from the Versions Directory list box to the Selected Files list box.

<< Clears all files from the Selected Files list box.

< Clears the highlighted file from the Selected Files list box.

Reference by

VERSIONS/VB assumes that you want to check out the most recent file version. To check out an older version, select either a specific **Version Label** or a specific **Version Number** from one of the list boxes.

To Change/To View

Sets lock status for the checked out files. Select **To Change** if you want to lock the files and make changes. Select **To View** if you do not plan to lock and make changes to the files.

Force Check-out

When checked, automatically replaces a newer file in your working directory with the older version selected from the versions directory. **Use with caution!** This option overwrites newer files in your working directory with older versions. To preserve the newer version of the file, check it in to Versions/VB before you overwrite it with the older file.

Suggest

VERSIONS/VB "suggests" which files to check in by entering changed files into the Selected Files list box automatically. If you have made changes and want to return to the VERSIONS/VB suggestions, click Suggest.

Note Visual Basic caches forms and modules if enough memory is available. This can lead to a minor problem if you check out a version of a form or module that is different from the one cached in memory. To refresh the cache, simply reopen the project using the Visual Basic **File => Open Project...** menu item. In all cases where you check out the current project .MAK file, be sure to reopen the project using the Visual Basic **File => Open Project...** menu item.

See Also

Check Out Hints
Why lock files?

Check In Hints

When should I check in files?

You should check in files when they are at a specific point of completion (for example, when you've finished creating all forms or when you've set all the context-sensitive help hooks).

For binary files, VERSIONS/VB stores only the number of versions you choose. When you reach the maximum number of versions, the oldest version is removed to make room for the new version, so you do not want to check in a file every time you make a minor change. Instead, keep the checked-out file in a working directory until you have made a significant revision.

Set the default maximum number of versions by selecting **User ⇨ Options** from the Project window menu bar.
Set the maximum number of versions for individual files by selecting **Versions ⇨ Maximum Versions** from the Project window menu bar.

See Also

Maximum # of Versions
Options

Note VERSIONS/VB will not allow you to check in a file that is locked by someone else. If you attempt to check in a file and see this message:

Not locked by you. Please lock the Versions file and retry operation.

Follow the steps below to resolve the problem:

1. Call the user who currently has the file locked and confirm that he or she has finished making changes. Verify that the most recent copy of the file has been checked in.
2. Request that unlock the file so you can have access to it. As soon as they unlock it, check out the file using the To Change (Lock) option. This puts your lock on the file.
3. When you complete your changes, check in the file.

Description/Reason for Check-in

Description

When you check in a file for the first time, you can enter a **Description** of up to 256 characters for the file. A file description is helpful in determining what a file does or how it is used in a project. File descriptions appear in the History and Project reports.

To change an existing file description:

1. Select **Versions** ➪ **Description** from the VERSIONS/VB menu bar.
2. Select the appropriate file from the Versions Directory list box. The existing description (if any) appears in the Description text box.
3. Edit the existing description, then click OK.

Reason for Check-in

When you check in existing files (files that have been checked in before) you can enter a **Reason for Check In** of up to 128 characters. Noting the reason for check-in is valuable for keeping track of changes made to the file.

Note Word-wrap is always enabled in the Description/Reason text box. If you want to save RETURN characters with your description, you must press the ENTER key. You can use the standard Windows cut, copy and paste keyboard shortcuts: CTRL-X, CTRL-C AND CTRL-V.

Maximum Number of Versions

You can change the maximum number of versions by entering the number into the Maximum # of Versions text box. The maximum number of versions of a file may be set to any value between 0 and 32767. A value of zero indicates that there is no limit to the number of versions retained. A typical value is ten. Note that blank is treated the same as zero.

In the following example, versions 1 and 2 are flagged as permanent **(BOLD)** while all subsequent versions are non-permanent. The maximum number of versions is set to two.

Version Checked in	**Versions retained**	**Comments**
1	**1**	Initial version marked permanent.
2	**1** **2**	Second version also marked permanent.
3	**1** **2** 3	Third version non-permanent.
4	**1** **2** 3 4	Fourth version, maximum versions reached.
5	**1** **2** 4 5	Fifth version, maximum exceeded so third (non-permanent) version deleted.
6	**1** **2** 5 6	Sixth version, maximum again exceeded so fourth (non-permanent) version deleted.

In this example, only two non-permanent versions are allowed. By the fifth version, version 3 has been eliminated. By the sixth version, both version 3 and 4 have been eliminated. Versions 1 and 2 are always retained, since they are permanent.

Check Out Hints

You'll want to check out files to:

- Update the files in your working directory
- Obtain the latest version of a file so you can work on it
- Retrieve a previous version of a file

If you are checking out one or more files to make changes to them, use the To Change (Lock) option to put your lock on the files. When you lock a file, VERSIONS/VB notifies other users that you are making changes to the file.

If you are checking out files for review only, do not lock them.

To retrieve a previous version of a file, select the version from the Check Out Wizard dialog (select **Versions ⇨ Check Out Wizard** from the Project window menu bar or click the Check Out Wizard button on the toolbar to display the Check Out Wizard dialog).

If you want to overwrite newer files in your working directory automatically, check the Force Check-out check box. **Use with caution!** This option overwrites newer files in your working directory with the older versions. To preserve the newer version of the file, check it in to VERSIONS/VB before you overwrite it with the older file.

See Also

Checking Out Files
Checking In Files

Check-In Wizard...

Opens the Check In Wizard dialog so that you can save the current version of your files in the versions directory.

This is the same as clicking the Check-In Wizard button on the VERSIONS/VB toolbar.

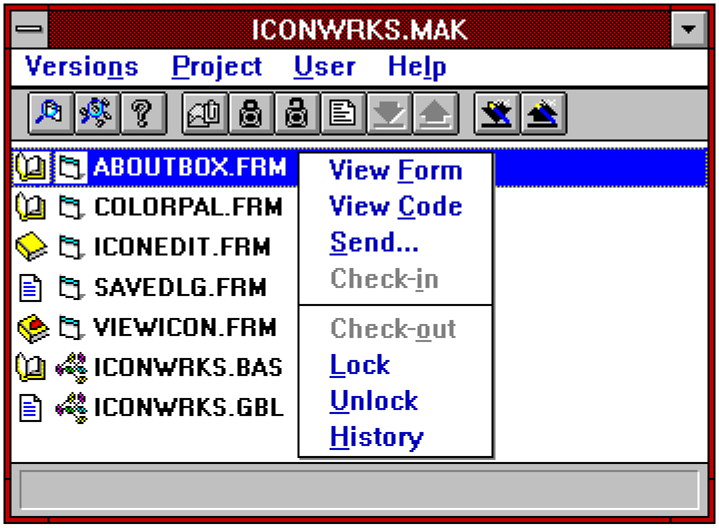
See Also

Checking In Files

VERSIONS/VB Project window

Starting VERSIONS/VB opens Visual Basic and the VERSIONS/VB Project window.

Click on each of the controls to view a brief description:



History

Produces a quick History report of the selected file using the current settings.

Unlock

Unlocks the selected file.

Lock

Assigns the selected file to you. Prevents others from checking in the file.

Check In

Performs a quick check-in of the selected file.

Performing a quick check-in retains file locks (or locks the file if this is a new file).

> **Note** Quick check-in is a shorthand method that is less flexible than using the Check-in Wizard >> button. You can also use the toolbar button to check in a single file.

Maximum Versions...

Opens the Maximum # of Versions dialog, which allows you to set the maximum number of versions for one or more files. This overrides the default maximum number of versions set for binary files in the Options dialog and is the only way to change the maximum number of versions for text files..

See Also

Maximum Number of Versions
Options

Status Line

When you access a drop-down menu, VERSIONS/VB displays a short description of the menu item.

When you click on a toolbar button, VERSIONS/VB displays a short description of its function.

When VERSIONS/VB is performing an operation like scanning the directory to retrieve the working files, it displays some useful information.

Working File List

Displays the project files in the .mak file and the Visual Basic file type bitmaps.

Optionally displays the VERSIONS/VB "book" bitmaps that indicate file status.

See Also

Visual File Attributes (books)

Deleting a File Version

You can delete any file version from a VERSIONS/VB project file unless it is the only version.

Dialog Box Options

Project
Displays the current project name and path.

Versions Directory
Project files contained in the versions directory appear in the list box. Select the file from which to delete a version.

Reference by
When you select a file, VERSIONS/VB displays the most recent version number. Choose the version to delete by selecting either a specific **Version Label** or a specific **Version Number** from one of the list boxes.

Note If the Confirm on Delete option is active, VERSIONS/VB asks for confirmation before it deletes a file version. If the status of the selected version is Permanent, VERSIONS/VB requests additional confirmation before deleting the version.

Use the Visual Basic **Remove File** menu option to remove a file from a Versions/VB project. Versions/VB asks whether you want to delete the file versions. Answer Yes to delete the version file from the version directory. If Confirm on Delete is active, Versions/VB requests confirmation before deleting the versions file. This process does not delete the working file.

See Also
Setting Options

Description

Add or change a file description.

Dialog Box Options

Project
Displays the current project name and path.

Versions Directory
Project files contained in the versions directory appear in the list box. Select a file to change its current description.

Description
Enter or edit a description (up to 256 characters) of the file.

Note If Sticky Dialogs are active, this dialog remains on the screen so you can view or change the description of more than one file.

Diff (Text Files Only)

Compare two different versions of the same text file.

Dialog Box Options

Project
Displays the current project name and path.

Versions Directory
Project files contained in the versions directory appear in the list box. Select a file to compare versions.

Reference by **Examples**

First Version
Select the first version of the file to compare.

Second Version
Select the second version of the file to compare.

Use External File Comparison Program
This check box is unavailable unless you have entered the name of an external comparison utility in the Options dialog. If available, check to use external comparison utility, uncheck to use the VERSIONS/VB file comparison utility, Visual Diff.

Edit Window

The Edit window displays file history reports, and project reports. A toolbar provides easy access to commonly used features.

The contents of the Edit window may be:

- **Printed**
  Select **File ➪ Print...** from the menu bar or press the print button to display the standard Windows print dialog.
- **Sent as mail**
  Select **File ➪ Send...** from the menu bar or press the Send button to display the Microsoft Mail Send dialog.
- **Edited in memory**
  Sections may be selected, cut, copied, pasted, and/or cleared. VERSIONS/VB uses the Windows clipboard for these functions, so you can copy or move part or all of the text displayed in the Edit window to another Windows application (for example, a word processing document).

The Edit window supports these standard Windows keyboard shortcuts for basic editing.

Undo	CTRL-Z
Cut	CTRL-X
Copy	CTRL-C
Paste	CTRL-V
Clear	DELETE

Mailing Working Files and Sending Notes

VERSIONS/VB allows you to access mail and attach any working file to a mail message.

To send a working file as an attachment:

1. Select the file in the VERSIONS/VB Project window.
2. Press the Send button >>.
   Or,
   Click the right mouse button to access the VERSIONS/VB pop-up menu and select **Send...**.

3. Follow the procedures for Microsoft Mail when sending any document via MAIL.

To send mail to another user:

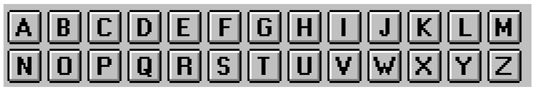
Select **User** ⇨ **Send Note to...** from the Project window menu bar.

Note The **Send...** menu option and the Send button are disabled unless a file is selected and mail is available.

See Also

FILENAME.MAK (Send option)

Glossary



A

Application

Archive

ASCII

ASCII file

Associations

B

Baseline

Bitmap

Button bar

C

Compression

Computer’s clock

Configuration Management

D

Delta

Difference log/report

Drag and drop

E

Embedded OLE object

Enhanced mode

F

Formal baseline

Compression

Compression is used to fit more data onto your hard disk. Compression effectively doubles your hard disk space using algorithms to squeeze data down to a minimum size. You can compress files on your hard disk using third-party compression utilities available.

We recommend use of compression to maximize hard disk use. The optimal solution for users is to use a third-party compression system (e.g., STACKER) widely available for your entire disk drive. This solution provides several advantages over having VERSIONS/VB perform compression:

- All files on your hard disk are compressed, not just versions files.
- Since these third party compression systems compress entire files, they would typically out perform any compression scheme we could provide on a version basis.
- If you already are using a third party compression system, performance would suffer if we also compressed individual versions (i.e., compression would occur twice).

We view compression as an operating system service benefiting all applications. In fact, many operating systems now provide compression.

Since most versions of files tend to be similar, we have found that compression ratios are a function of how many versions are kept for a file. For example, if you have a Versions file that contains 5 versions, you could get compression ratios of 5:1 or better. Note that these ratios apply to most file types, including Binary. Be aware that compression ratios vary depending on how much data is similar.

Application

A computer program designed to perform a specific task such as word processing or accounting.

Archive

A copy of a program or data file saved to ensure against loss in case the original materials are deleted or damaged. Archived files can be stored on tape, floppy disks, or another computer system.

ASCII

An acronym for **A**MERICAN **S**TANDARD **C**ODE FOR **I**NFORMATION **I**NTERCHANGE, pronounced “asskey”. A standard 7-bit standard code adopted to facilitate the interchange of data among various types of data processing and data communications equipment. This generic format is sometimes referred to as “text” and is useful for transferring unadorned but readable files between programs that could not otherwise understand each other’s files.

We recommend that you save your Visual Basic project files in text format. See Storage Strategies for more information.

ASCII file

A file stored in ASCII text format. An ASCII file contains characters, spaces, punctuation, carriage returns, and sometimes tabs and an end-of-file marker, but no formatting information. Also referred to as a text file, text-only file, or ASCII text file, an ASCII file can be edited and printed with a text editor or word processor.

We recommend that you save your Visual Basic project files in text format. See Storage Strategies for more information.

Associations

The assigning of specific filename extensions to applications using the Windows File Manager. The file and its associated program can be opened automatically when the you double-click on the file in the Windows File Manager.

Baseline

A common point (or base) that serves as a starting point. In a VERSIONS/VB project, a common starting point from which all individuals working on a project can obtain the most recent version of a file.

Binary file

A file containing a program or data in machine code. Examples of binary files and their respective file name extensions are executables (EXE), resources (RES), word processing documents (DOC - Word for Windows), spreadsheets (XLS - Excel), object files (OBJ) and bitmaps (BMP).

Bitmap

A graphical image stored in binary file format. Bitmap files are usually generated by a paint or screen capture program.

Toolbar

An interface element, usually found directly under the menu bar, that contains buttons providing quick access or shortcuts for commonly used menu items.

Configuration management

The process of managing the internal and external components and the software of a single computer or of a computer network.

Delta

A method of computing differences between progressive versions of a file.

With the Delta methodology, only the latest version of the file is saved in its entirety. In all other versions, only the differences between one version and the next are saved.

See Also

Version control system

Difference log/report

A list of differences between two versions of a text file.

Drag-and-drop

Moving a graphic image from one place to another on the screen using the mouse.

To accomplish a drag-and-drop operation in Windows:

1. Point to the object to be moved with the mouse pointer.
2. Hold down the left mouse button.
3. Drag the object to the new location by moving the mouse.
4. Release the left mouse button to complete the operation (drop the object being moved).

Enhanced mode

An operating mode for Windows that provides access to virtual memory capabilities of the 80386 processor.

INI File

INI files are startup files used by many Windows applications to record setup and customization information.

Interim versions

Any versions of a file between the first and the final version. Interim denotes that this file is still being modified and is not the final version.

Local drive

The disk drive located on your own local workstation rather than a drive located on a network or other remote location.

Stand-alone Workstation

Your local computer, which may or may not be connected to a network. This term is often used to distinguish between activities on your local computer system and activities on a network computer.

Locks

A software process applied to a file to prevent any unauthorized person from writing to it.

Mail

A program that enables users to send messages back and forth on a network.

Microsoft Mail

The mail program from Microsoft designed for Windows users. It is available through regular retail outlets and is part of the Microsoft Office suite of products as well as Windows for Workgroups.

See also

Mail

Milestone

A point denoting completion of a specific phase within the overall development of a project.

Non-permanent versions

Within VERSIONS/VB, if a version is not specifically defined as permanent, then it is non-permanent. The oldest non-permanent version is automatically deleted when the maximum number of versions is exceeded.

NOVELL

A leading local area network (LAN) manufacturer providing software and hardware designed to allow users to share files and other computer resources.

OLE

An acronym for **O**bject **L**inking and **E**mbedding, a method of transferring and sharing information among applications. Objects are items such as bitmaps, spreadsheets, or databases. Objects are either *linked* to a file or *embedded* in the file. If an object is linked, the file contains only a reference to the object. If an object is embedded in a document, the document actually contains the object.

Permanent versions

In VERSIONS/VB, this term designates files that are not implicitly deleted like temporary (non-permanent) versions. Permanent file versions can only be explicitly deleted by selecting the **Versions ➪ Delete...** menu option.

Project

An organized collection of the files used to complete an undertaking (for example, the files necessary to create a functioning icon editor, as in the Visual Basic ICONWRKS sample).

You must save a project in Visual Basic and designate a versions directory from within VERSIONS/VB before you can check in any files for version control.

Working Directory

The working directory stores the files that you work on day-to-day. It should be located wherever you perform your everyday tasks -- usually on your local drive, but may be on a network drive. VERSIONS/VB projects can have more than one working directory.

Versions Directory

The versions directory stores the special files that VERSIONS/VB uses to maintain copies of your work. It should be located wherever you store important copies of your work or share work with others -- usually on a network drive, but may be on your local drive. All VERSIONS/VB projects have a single versions directory.

Project Paths

Each project includes at least one working directory and only one versions directory. These two directories are the project paths in a VERSIONS/VB project.

Purging

The process of eliminating old or unneeded information.

StarBase Corporation

A software development corporation with offices in Irvine and Los Altos, California and Dallas, Texas. StarBase Corporation is a publicly traded company and is listed on the Vancouver Stock Exchange under the symbol **SBP**.

Status bar

A Windows interface element, used in many applications, that displays useful information, for example, whether Num lock and Caps lock are engaged, what the system is doing when busy (such as searching), help information for the currently selected menu option, and so on.

Version Label

A version label provides a convenient method of identifying a version of a file by name. This is valuable when you are checking in a group of files that may need to be checked out together. When multiple files are involved, each file will generally have a different number of versions. By selecting a version label, you can check out all files related to a specific change by referencing that version label.

In VERSIONS 1.1, this feature is called Symbolic Name.

See also

Version number

Technical Support

The process of helping customers resolve problems they encounter while using software products.

Version or Revision

The terms version and revision refer to a copy of a file saved in a version control file. One or more of these versions may be present in a version control file. In VERSIONS/VB all version control information is stored in versions directory files.

Version attributes

Each version of a file stored in a version control file has important information attached to it.

In VERSIONS/VB, the version attributes are:

- version number
- author
- date/time stamp
- size
- permanent or temporary
- Version Label (if any)
- Reason for check-in

Version control

Version control is the process of storing and tracking the various changes (versions) to a specific file. A version control system maintains the multiple versions generated as the file evolves into its final form.

The main advantage of using an automated version control system is fast, easy recall of previous versions. If you are on a network, VERSIONS/VB also provides a quick, easy-to-use method of obtaining versions of files being generated and maintained by other users working on a project with you.

Version number

The version number is an identification number assigned when a version is stored. This number is used when a you access a file by version number.

See also

Version label

Version control system

Application software to help manage multiple versions of the same file.

Version files

The version files contain all the file versions and are located in the versions directory. These files have the same name as the actual working files, except that they are written in a format unique to VERSIONS/VB These files cannot be read by the original applications, only by VERSIONS/VB Never directly edit or delete these files through Windows or DOS.

VERSIONSFULLNAME

VERSIONSFULLNAME is an environment variable used by VERSIONS/VB to identify who has a lock on a versions file and to send a mail message when you break another user's lock on a versions file. The VERSIONSFULLNAME environment variable is set in your autoexec.bat file at installation time.

Wildcard expression

Wildcard expressions are a simple form of regular expressions used to do pattern matching.

See your *MS-DOS user manual* for details on wildcard rules.

Wizard

A Wizard is a series of dialogs designed to walk first-time users through the setup of some object or process.

Working files

The working files are the actual files you are modifying. They are located in the working directory. This is in contrast to the version files which contain all the version information and are located in the versions directory.

Introduction

Today's professional Visual Basic developer needs an integrated, automated, easy-to-use, and error-free system for managing and tracking the many versions of the components they develop. VERSIONS/VB 1.1 provides the tightest and most user-friendly integration of version control for Visual Basic 3.0 available today. The Visual Basic project window is enhanced to include a custom Versions 1.1 menu, toolbar, pop-up menu and status line, which provide you with integrated access to all the version control functions you need to manage your Visual Basic project resources. It is network-compatible and especially well-suited to helping Visual Basic development teams collaborate through the use of:

- Point-and-click access to all integrated version control functions from the Visual Basic project window
- Check in of new form, module, VBX and DLL file versions
- Check out of form, module, VBX and DLL files from other team members
- Check out of specific form, module, VBX and DLL file versions
- Access on a specific milestone basis
- Locking of form, module, VBX and DLL files to prevent collisions between users
- Automatic access to FRM and FRX files as pairs
- Form, module, VBX and DLL file version history
- Visual side-by-side code comparison of any form, module, VBX and DLL file versions

Running under Microsoft Windows and fully integrated with Visual Basic, VERSIONS/VB:

- Provides full support for all binary file formats
- Provides efficient support for text (ASCII) files

  Stores only the differences between each version of a text file; the latest version is stored in its entirety
- Has no limit on the number or types of files that you can maintain
- Is fully compatible with VERSIONS 1.1 which may be used for all your general purpose version control needs

How does VERSIONS/VB work?

VERSIONS/VB tracks all changes to a Visual Basic project, either on a server or a local workstation, through your Visual Basic project. This project contains a versions directory used to store the versions of your form, module, VBX and DLL files and other important information, such as which users are currently working on specific files.

Users check form, module, VBX and DLL files into and out of the Visual Basic project versions directory as needed. VERSIONS/VB also provides assistance in suggesting which files need to be checked in or out by comparing files to determine which are newer. As users check in various project resources, VERSIONS/VB provides complete project reporting.

Maintaining previous versions

Since previous versions are kept in the Visual Basic project versions directory, you can retrieve them later, if needed. You can configure VERSIONS/VB to store versions in a number of different ways:

- Store all versions
- Store all versions until it reaches a user-defined maximum number of versions; older versions are then dropped
- Set a maximum number of versions and mark specific versions as permanent; permanent versions are always retained, regardless of the maximum number of versions set

Documentation

VERSIONS/VB allows you to maintain descriptions for each file and reasons why each file was checked in. This documentation is easily viewed through file history and project reporting.

To begin using VERSIONS/VB on a new or existing project see Getting Started.

Join Project

Allows you to join an existing VERSIONS/VB project.

Joining a VERSIONS/VB project is a two step process. First you must select a previously checked in Visual Basic project .MAK file. Second you must specify a new VERSIONS/VB project Working directory for this project.

Dialog Box Options

File Name
Displays the file search criteria until you select a file, then displays the selected file name.

List Files of Type
Allows you to change the file search criteria. The default is *.MAK.

Directories
Displays the current directory as you browse the displayed directory structure.

Drives
Select the drive on which to browse from the drop-down list box.

See Also
Creating Projects

Send

Allows you to prepare a mail message with the selected file automatically included as an attachment.

Help

Accesses online help.

Help Menu

Accesses various portions of VERSIONS/VB online help.

User Menu Close

Selections on the User menu are:

Options...
Sent Note to...

Project Menu

Close

Selections on the Projects menu are:

FILENAME.MAK
Options...

Join...
Versions Directory...

Reporting...
Milestone...

Refresh

Join...

Allows you to join a VERSIONS/VB project that already exists (has been checked in).

See Also

Join Project

FILENAME.MAK

Opens a cascading menu that allows you to perform common operations on the project .mak file, including:

Send...
Attaches the most recently saved project .mak file to a mail message.

Check-in
Checks the project .mak file in to the versions directory.

Check-out
Checks the most recent version of the project .mak file out of the versions directory.

Lock
Locks the project .mak file.

Unlock
Unlocks the project .mak file.

History
Generates a History report for the project .mak file using the current settings.

vbpath

Although it is not necessary to install Microsoft Visual Basic before you install Versions/VB, you must install Visual Basic before you launch Versions/VB.

If Visual Basic is installed on your system and Versions/VB still cannot locate it, edit your vb.ini file, located in your Windows directory, to add the line **`vbpath=c:\vb`** (substituting the location of Visual Basic on your system for the path shown here).

General Information

Lock

Locks the selected file.

Backups

It is extremely important that you maintain regular backups of all your files. This includes versions directory files stored either locally or on your network server. Please don't use VERSIONS/VB as a substitute for regular tape backups.

Locks

Lock or unlock a file. When you lock a file, VERSIONS/VB warns anyone who attempts to access the file that you are making changes. When you unlock a file, you allow other team members to lock and make changes to a file.

Dialog Box Options

Project
Displays the current project name and path.

Files in Versions Directory & Selected Files
Project files contained in the versions directory appear in the list box on the left. Enter files to be locked into the list box on the right.
⇨Copies the highlighted file from the Versions Directory list box to the Selected Files list box.
⇨Copies all files from the Versions Directory list box to the Selected Files list box.
⇨Clears all files from the Selected Files list box.
⇨Clears the highlighted file from the Selected Files list box.

Lock/Unlock
Sets lock status for the checked out files. Select **Lock** if you want to warn others that you own the files to make changes. Select **Unlock** when you are finished making changes to the files to release the files to other team members.

See Also
Why Lock Files?

Setting Project Options

Allows you to include .vbx files and .dll files as candidates for version control.

Dialog Box Options

VBX Files
Check to include .vbx files as candidates for version control.

DLL Files
Check to include .dll files as candidates for version control.

Note Project options affect the specific project in which they are set on the specific workstation for which they are set. For shared projects, be sure that each team member uses the same settings.

Why Lock Files?

Locking files is important in a workgroup or network environment. It is a protective measure to keep other users from overwriting your work.

For example:

A form is stored on the network server as part of a VERSIONS/VB project. You want to work on this form, so you check it out of the versions directory and lock it. VERSIONS/VB copies the most recent version of the file into your working directory (which is located on your local disk) and you begin to make changes.

Another team member attempts to check out and lock the file. Because you already have the file locked, VERSIONS/VB:

- Warns that user that the file is locked by you
- Prevents that user from checking in the file without notifying you (if you are using Microsoft Mail)

If you need to access a file that is locked by someone else:

1. Contact the current file owner to see if changes to the file have been completed. Ask the user to check in and unlock the file.
2. Once the file is released, check it out of the versions directory using the To Change (Lock) option.
3. When you complete your changes, check in and unlock the file.

Note If you forcibly break another user's lock, VERSIONS/VB notifies that user by sending a mail message (if you are using Microsoft Mail).

Join Project - Working Directory

Allows you to defined the Working directory for a newly joined Visual Basic project .MAK file.

Dialog Box Options

Directories
Displays the current directory as you browse the displayed directory structure.

Directory list
Displays your system's directory structure. Double-click on drive and directory names to build the path in the text box above.

Drives
Select the drive on which to browse from the drop-down list box.

New Directory
Allows you to create a new directory without exiting from the dialog. To create a new directory:

1. Navigate the directory structure above until you reach the place where the new directory will be located.

2. Type the new directory name in the text box.

3. Click the Create button.

The Create button is disabled if the New Directory text box :

- is empty
- contains an invalid directory name
- contains a directory name that already exists in the current directory

See Also
Creating Projects

Maximum Number of Versions

Change the maximum number of non-permanent versions to be retained for a file. Permanent versions are always retained.

Dialog Box Options

Project
Displays the current project name and path.

Files in Versions Directory & Selected Files
Project files contained in the versions directory appear in the list box on the left. Enter files for which the maximum number of versions is to change into the list box on the right.

➪Copies the highlighted file from the Versions Directory list box to the Selected Files list box.

➪Copies all files from the Versions Directory list box to the Selected Files list box.

➪Clears all files from the Selected Files list box.

➪Clears the highlighted file from the Selected Files list box.

Max # of Versions
Enter the new maximum number of versions.

Note Limiting the number of retained versions allows VERSIONS/VB to run more efficiently and use less disk space. The more versions of a file you retain, the slower VERSIONS/VB runs and the more disk space it uses.

See Also
Max Versions

Visual File Attributes (Books)

When **Show Books** is checked, VERSIONS/VB displays a bitmap representing the current state of the file next to the filename in the Project window:

🗎

File has never been checked in.

📒

File has been previously checked in. No one currently has the file locked.

📖

File has been previously checked in. You currently have the file locked.

📕

File has been previously checked in. Another user currently has the file locked.

Permanent...

Opens the Permanent/Non-Permanent dialog, which allows you to set permanent or non-permanent status for one or more files.

See Also

Permanent

Milestone

Use when you reach a project milestone event to create a set of permanent file versions, and optionally, store a set of project files within a subdirectory especially created for this purpose. You must have all files locked to create a milestone.

Dialog Box Options

Project
Displays the current project name and path.

Milestone Name
Type a unique name for the milestone (a name that has not previously been used for a milestone or version label). This name will appear in the Reason for Check In for all file versions included in the milestone.

Archive
Check to copy a read-only set of milestone files to a separate subdirectory.

Compress
Check to compress the archive files. This option assumes that compress.exe is available on your system in a directory referenced by the PATH statement in your autoexec.bat file.

Archive Name
Type the name of the Archive directory to be created below the versions directory. (Follow standard MS-DOS directory naming conventions.)

Note Creating a milestone does not involve any files in your working directory. Only the latest versions of files checked in to the versions directory are used to create a milestone.

See Also
Checking out files
Permanent/Non-Permanent versions
Deleting a version

Description...

Opens the Description dialog, which allows you to change the description for one or more files.

See Also
Description

Delete...

Allows you to delete one or more file versions from a VERSIONS/VB file.

See Also

Deleting a File Version

Versions Directory...

Allows you to specify a versions directory for the current Visual Basic project.

See Also

Setting the Versions Directory

History

Produces a quick History Report of the selected file using the current settings.

Quick Check In

Performs a quick check-in of the currently selected file.

This toolbar button is enabled only if the file needs to be checked in (has been modified since the last time you checked it in or has never been checked in before).

Performing a quick check-in retains file locks (or locks the file if this is a new file).

Note Quick check-in is a shorthand method that is less flexible than using the Check-in Wizard ➪ button. You can also use the pop-up menu to check in a single file.

Messages

Archive <ArchiveName> already exists. You must enter a new archive name.

When archiving a project through Project milestone, you must enter an Archive name that has not been previously used in the Archive Name control.

Can't copy <FileName> to archive.

An error occurred while trying to copy a file to a Project milestone archive.

Can't create archive directory <ArchiveName>.

VERSIONS/VB encountered an error while trying to create the directory <ArchiveName> while creating a project milestone. Make sure you have write privileges to the VERSIONS/VB versions directory and attempt the operation again.

Can't create Versions Directory <VersionsDirectory>.

VERSIONS/VB encountered an error while trying to create the Versions Directory <VersionsDirectory>. Make sure you have write privileges to the drive and attempt the operation again.

Can't create Working directory <WorkingDirectory>.

VERSIONS/VB encountered an error while trying to create the Working directory <WorkingDirectory>. Make sure you have write privileges to the drive and attempt the operation again.

Can't delete only version in Versions file.

You have attempted to delete the only version in a Versions file. This is not allowed. At least one version must always be present in a Versions file. If you want to delete all versions in a Versions file, you must delete the Versions file using the Visual Basic **Remove File** option.

Can't diff same versions.

You have attempted to diff the same version of a Versions file. Select different First Version and Second Version numbers.

Can't find <COMPRESS.EXE> on your MS-DOS path. You can't compress an archive without it.

See the on-line help for more information.

While attempting to archive a milestone, Versions/VB could not find COMPRESS.EXE in your PATH. Add the directory containing COMPRESS.EXE to your path.

Can't find symbolic name.

VERSIONS/VB was unable to access a version using the version label you supplied.

Can't find version number.

VERSIONS/VB was unable to access a version using the version number you supplied.

Can't generate report, projrpt.txt already exists in project.

While attempting to archive a milestone, VERSIONS/VB attempted to generate a project report named projrpt.txt. However, a file with this name already existed. Generation of the project report is suppressed.

Can't log on to mail as <VERSIONSFULLNAME>.

VERSIONS/VB was unable to log you into Microsoft Mail for Windows. Try logging in again with a valid logon name and password.

Can't run external diff command <ExternalDiffCommand>.

VERSIONS/VB was unable to run external Diff command you entered into the Option dialog. Verify that the command you entered is correct and references an executable file that exists.

Can't run Visual Diff.

VERSIONS/VB was unable to run Visual Diff. Verify that VISDIFF.EXE is installed in the same directory as VERSIONS/VB.

Edit control out of space.

The Edit dialog used for Project Reporting, History, etc., contains a multi-line-edit control that is limited to 55,534 bytes of data. However, the effective limit is actually somewhat smaller and will vary from machine to machine.

File <FileName> too large, loading first 55535 bytes.

The Edit dialog supports files up to 55,535 bytes in length. If the file being loaded is longer than this maximum limit, only the first 55,535 bytes are used.

Versions Directory <DirectoryName> no longer exists.

VERSIONS/VB can't find the Versions Directory in <*ProjectName*>. If you deleted the Versions Directory with the Windows File Manager, then you will probably want to delete this project. If you don't think this error should be occurring, check to be sure you have the correct network connections defined for the drive that holds the Versions Directory. Either delete the project or connect to the network drive that holds the Versions directory.

Invalid Working directory found in project <ProjectName>.

VERSIONS/VB can't find the Working directory in <*ProjectName*>. If you deleted the Working directory with the Windows File Manager, then you will probably want to delete this project. If you don't think this error should be occurring, check to be sure you have the correct network connections defined for the drive that holds the Working directory. Either delete the project or connect to the network drive that holds the Working directory.

Locked by another user

You have attempted to check out or lock a file that is currently locked by another user. You may break the user's lock (however, the user will be notified by mail), and retry the operation.

Not locked by you. Please lock the Versions file and retry operation.

You have attempted to perform some action that requires a lock that you do not have on the file being accessed. Lock the file.

System Error: Can't copy <FileName> to archive.

VERSIONS/VB encountered an error while trying to copy <FileName> to the milestone archive. Try closing other applications that might have <FileName> open and attempt the operation again.

System Error: Can't create network lock file. You must have write privilege on Versions drive.

VERSIONS/VB was unable to create a necessary semaphore lock file. This can happen if you don't have write privileges to the Versions directory since all semaphore lock files are created there. Make sure you have write access to the Versions directory.

System Error: Can't create temporary file name

VERSIONS/VB was unable to create a necessary temporary file. This can happen if you don't have write privileges to the Versions directory since all temporary files are created there. Make sure you have write access to the Versions directory..

System Error: Can't open file <FileName>.

VERSIONS/VB could not open <FileName>. Try closing other applications that might have <FileName> open and attempt the operation again.

System Error: Can't open report file.

VERSIONS/VB was unable to create either a History or Project report. Check to see if your disk is full and, if so, free up some space and retry.

System Error: Can't open temporary file

VERSIONS/VB was unable to open a necessary temporary file.

System Error: Can't open Versions file

You have attempted to access a Versions file that is currently open in some other application. From the other application, close the Versions file and retry this operation. Another possible cause of this error is that some other process has deleted the file just prior to your attempt to access it.

System Error: Can't open working file. Another application may already have it open.

You have attempted to access a working file that is currently open in some other application. From the other application, close the working file and retry this operation.

System Error: Can't reconstruct ASCII file.

The Versions file has been corrupted. Restore a valid copy from a backup.

System Error: Can't set environment variable

VERSIONS/VB always reassigns the TMP environment variable to the Versions directory for temporary file creation. This is done to improve performance (on networks).

System Error: Incorrect Versions file format

You attempted to access a file in a Versions directory that is either corrupted or not a Versions file. Versions directories must contain only Versions files. If you have files of some other type in a Versions directory, remove them. If you're sure that you have only Versions files in your Versions directory, and you get this error, chances are that the file has somehow become corrupted and will need to be restored from some backup medium.

System Error: Can't allocate memory for diff.

System Error: Can't allocate memory for file <FileName>.

System Error: No more memory

Out of memory

Versions/VB ran out of memory while attempting to allocate Windows memory. You can try one of the following:

1. Close other running applications and retry operation.
2. Increase your permanent swap space from the Windows Control Panel and retry operation.
3. Install more memory and retry operation.

System Error: Read error

Versions/VB was unable to read the correct number of bytes while reading from a Versions file. This usually means that the file is not a Versions file or is corrupt.

System Error: Write error

Versions/VB was unable to write the correct number of bytes while writing to a Versions file or Working file. Check to see if your disk is full.

Symbolic name already used in previous version

You have attempted to assign a version label that has already been used in a previous version. Use another version label.

Unable to send document.

Check within the mail system to verify that you have the correct UserName. If so, contact the mail administrator for your network.

Unable to send mail re:broken lock to <UserName>.

VERSIONS/VB was unable to send mail to <UserName> informing them that you have broken their lock. This error occurs if <UserName> does not have a mail account.

Versions directory must not be root: <VersionsDirectory>.

The Versions directory must be a subdirectory, e.g., F:\SAM instead of F:\.

Versions file is in use by another user

The file you are trying to access is in use by another VERSIONS/VB user. Press the retry button to try again. VERSIONS/VB locks files on a network while performing operations. These locks are usually around for very short periods of time and this message is unusual.

Versions file <VersionsFile> already locked by <UserName>, do you want to break that lock?

The versions file <VersionsFile> is currently locked by <UserName>. If you click YES, that lock will be broken and a mail message will be sent to <UserName> informing them of your action.

Versions file missing

The versions file is missing from the project's Versions directory. Check to make sure it has not been deleted.

Versions file already exists

The file you are attempting to check in for the very first time has already been checked in. The only way this can happen is if some other user just happened to check in a file with the same name just prior to your attempt.

Working and Versions directories must be different

When creating or changing a project, you defined a Working directory and Versions directory that were the same. These two paths *must* be different. Define different Working and Versions directories.

Working directory must not be root: <WorkingDirectory>.

The Working directory must be a subdirectory, e.g., C:\SAM instead of C:\.

Working file missing

The Working file is missing from the project's Working directory. Check to make sure it has not been deleted.

Working file is not newer than latest version

You have attempted to check in a Working file with a date and time stamp that is older than the latest version in the Versions file. This is never allowed. Common situations where this can occur are:

1. Someone has checked in a newer version of the file. In this case, you will need to save your working file to some temporary place, check out the latest version, merge your working file changes with the latest version checked out, and then check in this merged working file.

2. You have checked out an older version of a file without modifying it and are attempting to check it in again. This is not allowed. If you want to check in the file you will need to edit it, save it (to update its date and time stamp), and then check it in.

3. You are accidentally trying to check in an old version of a file. Do not check it in. Your working file is not up-to-date with the project.You probably need to check out the file in question.

Working file newer than requested version

You have attempted to check out a version over a Working file with a newer date and time stamp. Remove the Working file or check the *Force check out* control in the Check out dialog. **Use caution** when doing this so as not to delete any work in progress.

Working file same as latest version, date and time stamp of working file set to latest version

You attempted to check in a file that is not different from the latest version currently checked in. VERSIONS/VB re-assigns the date and time stamp of the Working file with that of the latest (same) version.

You do not have a lock on one or more files in project <ProjectName>.

When archiving a project through Project milestone, you must have all of the files locked. Lock all files in the project and retry.

You have selected more than one file and a version number that may not be available in all files.

You are checking out more than one file and explicitly choosing a version number. There may be one or more files that do not contain this version. If so, and you choose to continue, you will receive warnings for those files. Normally, you should only specify an explicit version number when checking out a single file.

You must define the environment variable VERSIONSFULLNAME before starting VERSIONS/VB.

The environment variable VERSIONSFULLNAME must be defined in your AUTOEXEC.BAT file. VERSIONS/VB uses this for mail, locking, and other administrative tasks. See the installation instructions in your VERSIONS/VB user manual for more information on defining this variable.

VERSIONS/VB does not support files contained in root directories. Version control disabled for <Filename>.

You have included a file stored in your root directory in your VERSIONS/VB project. This file will not be a candidate for version control. To enable version control for the file:

1. Use the Visual Basic **File => Remove File** menu option to drop the file from the project.

2. Move the file to a subdirectory.

3. Use the Visual Basic **File => Add File** menu option to add the file back into the project.

Please rename duplicate files and reopen the project

You have attempted to access an existing Visual Basic project in which files with duplicate file names have been included using the Visual Basic **File => Add File** menu option. VERSIONS/VB cannot accept projects with duplicate filenames. You must rename duplicate filenames before you can open the project using VERSIONS/VB.

Form or code module name <Filename> already exists in this project. Please rename one of these files, then reopen the project.

You have used the Visual Basic **File => Add File** menu option to add files to the project that duplicate filenames already in use. This is not allowed.

You have checked out another version of project file <Filename>. Would you like to reopen the project now?

When you check out a file version, you must reopen the project before you can view the changes.

Versioned project file <Filename> and its associated files successfully checked out into working directory <DirectoryName>. Project must be reopened to review new file versions. Would you like to reopen the project now?

You have joined a VERSIONS/VB project and must open the project before you can work on the project.

Project file <Filename> is not a Versions file.

You have attempted to join a VERSIONS/VB project and the .mak file you selected is not a versioned file. Retry the operation and specify a .mak file stored in a versions directory.

Working directory <DirectoryName> does not exist. Create it now?

You have attempted to check out a file included in the project using the Visual Basic **File => Add File** menu option. Added files are referenced in the project .mak file by absolute path and this path does not exist on your system. VERSIONS/VB will create the necessary directory structure and check out the file.

Please shut down Visual Basic, then restart Versions/VB.

You have attempted to launch VERSIONS/VB while Visual Basic is already running. When you launch VERSIONS/VB, Visual Basic is launched automatically. Be sure that Visual Basic is not running, then launch VERSIONS/VB.

Versions/VB is already active.

You can only run one instance of VERSIONS/VB at a time on any given machine.

Visual Basic could not be activated.

VERSIONS/VB could not launch Visual Basic. Be sure that Visual Basic is properly installed on your machine. If this message still appears, edit your vb.ini file, located in your Windows directory, to add the line **vbpath=c:\vb** (substituting the location of Visual Basic on your system for the path shown here).

Visual Basic is still active. Close Versions/VB anyway?

You have attempted to close VERSIONS/VB while Visual Basic is still running. The Project window will return to the standard Visual Basic configuration and version control will no longer be available.

You do not have the latest version of <Filename> in your Working directory. Would you like to check out the latest version with a lock at this time?

You have attempted to lock a working file that is older than the most recently checked in version.

Performance Tuning Tips

VERSIONS/VB Project window:

You can configure the VERSIONS/VB Project window to show visual file attributes (books) that indicate the status of each file. When you elect to Show Books, VERSIONS/VB scans the state of each file and stores it in cache memory during startup and whenever you change projects. Storing this information in cache memory speeds up the process of displaying the Project window file list.

If you do not want to wait for VERSIONS/VB to scan the state of each file, or if you are working on a very slow network, uncheck Show Books in the Options dialog, This speeds up painting the Project window file list and allows you to move from one project to another more quickly. The Project window file list will display only filenames and Visual Basic bitmaps.

File Storage Management:

Limit the maximum number of versions of application files to some reasonable value — that is, the minimum number of versions required for a specific file. If you only need to keep three versions of a file, set the maximum number of versions to three. The more versions you keep, the longer it takes VERSIONS/VB to perform actions such as check-in.

VERSIONS/VB ships with a default maximum number of versions set to 10.

- You can change the default maximum number of versions in the Options dialog.
- You can change the maximum number of versions for a specific file when you check in that file for the first time.
- You can change the maximum number of versions for any file at any time by selecting the **Versions => Maximum Versions** menu option.

You can also reclaim disk space and speed up VERSIONS/VB by periodically purging your Versions files.

Learning Versions/VB

A special Quick Start tutorial in the front of your VERSIONS/VB user manual offers a step-by-step tour of the main features of VERSIONS/VB, including:

- Installing VERSIONS/VB.
- Creating a VERSIONS/VB project from existing Visual Basic files.
- Checking in files.
- Checking out files.
- Changing file descriptions.
- Viewing file history.
- Comparing two versions of a file.
- Creating a permanent version.
- Deleting a file version.

Your Computer's Clock

Each time you save a file version in the versions directory, VERSIONS/VB stamps it with the date and time from your computer's built-in clock. VERSIONS/VB determines which files need to be checked in or checked out by comparing the date and time stamp of your Working files against corresponding versions directory files.

If your computer’s clock is not correct or if it is not synchronized with every other project member’s clock, VERSIONS/VB will record the wrong date and time. This means VERSIONS/VB won’t be able to accurately determine whether the file needs to be checked in or checked out.

If you are working on a network, we recommend that you use whatever service your network server supports for synchronizing all workstation clocks.

See Also

Running VERSIONS/VB with Windows NT 3.1

Acknowledgments

The following people contributed their time and efforts in helping to bring this product to completion:

Alan Kucheck
Amanda Humphreys
Bill Stow
Brian Hess
Bryan Flynn
Christopher Shapley
David Mackinder
Jack Brantley
John Hawkins
John Juback
Jon Marinello
Julie Gray
Lionel Lavallee
Martha Dudley
Matt Birder
Naoma Stow
Sandi Byars
Steve Reynolds
Steven Yamanaka
Sunny Wong
Susan Gallagher

and Mike Benson

Microsoft Mail for Windows

If Microsoft Mail for Windows (available from Microsoft) is available at your installation, VERSIONS/VB will access it.

- When you forcibly break another user’s lock on a file to lock it yourself, VERSIONS/VB sends mail to inform that user of your action.
- You can send any working file or the contents of the Edit window to another user as an attachment.
- You can send notes to other users.

If Microsoft Mail for Windows is not available to VERSIONS/VB, all mail functions in VERSIONS/VB are disabled.

Whether Microsoft Mail for Windows is available or not, be sure to set the VERSIONSFULLNAME environment variable in your autoexec.bat file. (See the installation instructions in your VERSIONS/VB User Manual for specific instructions.)

See Also

Mailing Working Files

➪

StarBase Technical Support

If you have a question about VERSIONS/VB, you can:

1. Consult the on-line help and written documentation.

2. See the README.TXT file that came with VERSIONS/VB.

3. Post a message to our CompuServe forum under the Versions section. Log on to CompuServe and type **GO STARBASE**.

If you can't find the answer from any of these sources, or if you require information about product training, registration, or replacement disks, call StarBase Technical Support Services at (714) 442-4460 between 9 A.M. and 5 P.M. Pacific time, Monday through Friday.

When you call StarBase Corporation Technical Support Services, be prepared to give the following:

- The version number of VERSIONS/VB software you are using.

- The exact error message you received, if any.

- A precise explanation of what you were doing when the error occurred.

- The type of hardware you are using.

- The network operating system you are using, if any.

StarBase support services are subject to StarBase prices, terms, and conditions in place at the time the service is used.

StarBase also offers VERSIONS/VB training programs that can be tailored to your needs.

For information about product upgrades and pricing, call StarBase Corporation at (714) 442-4400 between 9 A.M. and 5 P.M. Pacific time.

Limitations

- Only files that are allowed in a Visual Basic project (forms, modules, VBX and DLL files) are candidates for version control. This is because VERSIONS/VB uses the Visual Basic project file as the VERSIONS project definition. If you want to do version control of other files (project plans, designs, online help, typeset documentation, tests, etc.) you may purchase VERSIONS 1.1 separately from StarBase Corporation. Call us at (714) 442-4400.

- Visual Basic caches forms and modules if enough memory is available. This can lead to a minor problem if you check out a version of a form or module that is different from the one cached in memory. To refresh the cache, simply reopen the project using the Visual Basic **File => Open Project...** menu item.

- The Working and Versions directories must be a subdirectory, e.g., C:\MYPROJ instead of C:\. All Visual Basic files that are added to a VERSIONS/VB project must also be stored in subdirectories.

- Only one instance of VERSIONS/VB may run at a time on any given machine.

- The Edit window (used for History, and Project Reporting) is limited to 55,534 bytes of data. However, the effective limit is actually somewhat smaller and varies from machine to machine.

Copyright Notice

Sticky Dialogs

Many of the tasks you perform in VERSIONS/VB are repetitive.

For example:

If you need to enter descriptions for 10 files, you don’t want to have to select **Versions ⇨ Description...** from the menu bar 10 times. It's easier to select a file, enter a description, click OK, select another file, enter a description, and so on.
By activating Sticky Dialogs, many dialog boxes remain active until you click Cancel.
The Sticky Dialogs option affects several dialogs, including Check In, Check Out, History, Locks, and Description.
To activate Sticky Dialogs:

1. Select **User ⇨ Options** from the VERSIONS/VB menu bar.
2. Check the Sticky Dialogs check box.

Organizing your VERSIONS/VB directories

Each project you create in VERSIONS/VB must consist of at least two directories:

- At least one **working directory** in which you'll keep the files you work on from day to day. The working directory should be wherever you keep your daily work — either on your local drive or on the network.

- A **versions directory** in which VERSIONS/VB stores:
  - As many temporary versions of your files as you choose to keep. By default, VERSIONS/VB keeps 10 temporary versions of your binary project files and unlimited versions of your project text files.
    See Options to change the default number of versions for new binary files.
    See Versions => Maximum Versions to change the number of versions for any VERSIONS/VB file (text or binary).
  - All the permanent versions of your files that you create.
    See Versions => Permanent to create a permanent file version.
  - Information about each file and file version, such as the file description, the reason why each version was checked in, and who, if anyone, is currently making changes to the file.

  If you're working on a stand-alone project, the versions directory may be on your local disk or in your logon directory on the network. If you're sharing files with others, the versions directory should be located where it can be accessed by everyone on the project.

Once you define the project, you no longer have to worry about the directory locations. You need only remember the name of the project.

If your project contains files included in the project using the Visual Basic **Add Files** option, we strongly recommend that all team members create identical directory structures so that the included files will be updated appropriately for all projects.

VERSIONS/VB Storage Strategy

When you use VERSIONS/VB to manage project files stored in binary format, VERSIONS/VB stores a complete copy of each version of your file.

This storage approach gives you fast access to each version of your file; however, the more versions of a file you store, the more disk space VERSIONS/VB uses for your versions directory.

We recommend the following guidelines:

- Do not check in a file each time you make a minor change. Wait until there have been significant revisions to the file. For example, check in a file when you complete all dialog boxes or set all context-sensitive help hooks. Remember, when you reach the maximum number of versions for the file, VERSIONS/VB deletes older versions to make room for newer versions.

- Give careful consideration to the number of versions of each file you maintain. The default number of versions for binary files is 10, but you can change this to customize the number of versions for each of your project files, if necessary.

- Create permanent versions only for significant milestones in your project. For example, create a permanent version of project files at alpha release, beta release, and first customer ship..

- Purge your versions directory on a schedule, such as each time a project milestone is reached. Purging the versions directory retains all permanent versions and the most current version of a file, and deletes all interim versions.

When you use VERSIONS/VB to manage text (ASCII) files, VERSIONS/VB stores your files using "Reverse Deltas". This means that only the most recent version of your file is stored in its entirety. For previous versions of text files, VERSIONS/VB only stores the differences or "deltas", so all versions of an text file can be stored with minimum impact on disk space.

When you check in a text file, VERSIONS/VB sets the maximum number of versions to unlimited (represented by blank or zero). You can change this to retain fewer versions using the Maximum Versions dialog and you can still create permanent versions and perform versions directory purges.

We recommend that you save all your Visual Basic files in text format to conserve disk space and to take advantage of the file comparison capabilities of Visual Diff.

cache

A special memory subsystem that stores frequently used information for fast retrieval.

Options

Set the behavior and appearance of VERSIONS/VB to suit your personal needs. These settings are saved from session to session.

Dialog Box Options

Edit Window

Governs the appearance of the VERSIONS/VB Edit window. This window displays History and Project reports.

Tab Stops

Sets the position of tab stops. The default of 8 specifies 8 blank spaces between tab stops.

Font

Opens the standard Windows Fonts dialog so you can specify a display font for the window. We recommend that you use a fixed-space font such as Courier for the Edit window display.

Confirmations

Specifies when VERSIONS/VB displays confirmation messages.

Confirm on Delete

When checked, VERSIONS/VB asks for confirmation when you delete a version or remove a file.

Confirm on Error

When checked, VERSIONS/VB pauses to inform you of an error in processing, such as when you attempt to check-in a file locked by another user.

Check-in Wizard on Exit

When checked, VERSIONS/VB displays the Check-in Wizard, if necessary, when you exit..

Other

Controls miscellaneous options.

Default Max Versions

Sets the number of maximum versions automatically assigned to all project binary files. You can override this setting for individual files by selecting the **Versions ➪ Maximum Versions** menu option.

External File Comparison Program

Specifies an external utility to use when comparing text files. If none is specified, VERSIONS/VB uses its own comparison utility.

Sticky Dialogs

When checked, alters the behavior of the OK and Cancel buttons on VERSIONS/VB dialogs. When Sticky Dialogs are in effect, clicking OK begins the specified operation but leaves the dialog open so that you can perform the same operation again on a different set of files. Clicking Cancel closes the dialog when you are finished using it.

Show Books

Turns the visual file attributes (books) display on and off.

Note **Default Max Versions** governs the number of binary files that VERSIONS/VB stores. Because of the storage strategies VERSIONS/VB uses, the maximum number of versions for text (ASCII) files always defaults to unlimited, which is represented by 0 (zero). You can change this in the Maximum # of Versions dialog.

See Also

Project Options

Quick Check Out

Performs a quick check out of the currently selected file.

This toolbar button is enabled only if a file is selected, that file has been checked in by someone else, and your working copy is out-of-date.

Note Quick check-out is a shorthand method that is less flexible than using the Check-out Wizard button . You can also use the pop-up menu to check out a single file.

Check In Wizard

Opens the Check-In Wizard dialog so you can check in one or more files.

Check Out Wizard

Opens the Check-Out Wizard dialog so you can check out one or more files.

Pop-Up Menu Close

The VERSIONS/VB pop-up menu provides quick access to the most common operations you'll perform on individual files.

To access the pop-up menu:

1. Point to the file to be acted on.

2. Click the right mouse button.

Selections on the pop-up menu are:

View Form
View Code
Send...
Check In

Check Out
Lock
Unlock
History

Purge...

Opens the Purge dialog, which allows you to delete all but the Permanent and most recent versions of a file.

See Also

Purging Files

Getting Started

Getting VERSIONS/VB Up and Running

Make sure you have completed these steps:

- Run the setup program to install VERSIONS/VB on your hard disk
- Change your autoexec.bat file to set VERSIONSFULLNAME=name

Consult your VERSIONS/VB User Manual if you need help with any of the above steps.

Using VERSIONS/VB

- We recommend that you work through the special Quick Start tutorial in the front of your Versions/VB user manual.
- You can create a VERSIONS/VB project from any Visual Basic project.
  - Open an existing Visual Basic project or create and save a new Visual Basic project.
  - Designate a versions directory in which VERSIONS/VB will store your versions files.
- You can join any VERSIONS/VB project already in progress.
  - Access the VERSIONS/VB Join dialog and specify the .mak file in the versions directory for the project you want to join.

See Also
Creating Projects

Unlock

Unlocks the selected file.

Permanent

Sets or removes Permanent status for a file. Permanent versions are not deleted from the system when you reach the maximum number of versions or when you purge files.

Since permanent versions affect system performance and increase disk space demands, only assign permanent status to very important versions of a file.

Dialog Box Options

Project
Displays the current project name and path.

Files in Versions Directory & Selected Files
Project files contained in the versions directory appear in the list box on the left. Enter files for which the status is to be changed into the Selected Files list box.
⇨Copies the highlighted file from the Versions Directory list box to the Selected Files list box.
⇨Copies all files from the Versions Directory list box to the Selected Files list box.
⇨Clears all files from the Selected Files list box.
⇨Clears the highlighted file from the Selected Files list box.

Reference by
VERSIONS/VB assumes that you want to change the status of the most recent file version. To change the status of an older version, select either a specific **Version Label** or a specific **Version Number** from one of the list boxes.

See Also
Maximum Number of Versions
Milestone

Using VERSIONS/VB and Visual Basic

Starting VERSIONS/VB

VERSIONS/VB accepts any command line argument that is valid for use with Visual Basic.

VERSIONS/VB can be used with any other Visual Basic add-on program (for example, VBAssist). To launch a Visual Basic add-on, edit the VERSIONS/VB ini file, versvb.ini, which is located in your Windows directory, and add the LAUNCHPROGRAM instruction.

For example, the following code will launch VBAssist and Visual Basic when you launch VERSIONS/VB:

```
LAUNCHPROGRAM=C:\VBASSIST\VBASSIST.EXE
```

Be sure to specify the full path to the Visual Basic add-on program.

When you launch VERSIONS/VB, you also launch Visual Basic. VERSIONS/VB enhances the Visual Basic Project window, replacing the Visual Basic View Form and View Code buttons with the VERSIONS/VB menu and toolbar.

Exiting from VERSIONS/VB

When you exit from Visual Basic, VERSIONS/VB terminates automatically.

You may also close VERSIONS/VB by selecting Close from the control menu associated with the VERSIONS/VB icon, which remains minimized on the desktop while you work with VERSIONS/VB and Visual Basic. When you terminate VERSIONS/VB in this manner, VERSIONS/VB returns the Visual Basic Project window to its normal state.

See Also

vbpath

Why Purge Files?

As work progresses on a project, you check more and more file versions into the VERSIONS/VB versions directory. Many of these versions represent work in progress and can be purged, or deleted, from the system when the project reaches a completion point (e.g., first customer ship). Purging files reduces the amount of disk space used by VERSIONS/VB files.

When you purge a VERSIONS/VB file, all permanent versions are retained and the latest version, whether permanent or not, is always retained. All other versions are deleted from the versions directory and from the disk on which they are stored. **Use with extreme caution**.

Project Reporting

Produce a project report for the selected project A project report displays detail information about all the files stored in the versions directory.

Dialog Box Options

Project
Displays the current project name and path.

Report on File Attributes
Select the attributes to be included in the report by activating the appropriate check box. By default, all attributes are included.

See also
Viewing File History

Diff (Text Files Only)...

Opens the Diff dialog, which allows you to select two versions of a text file and view the differences between them.

See Also

Diff

History...

Opens the History dialog, which allows you to specify attributes for, and generate a report of, all versions of the selected file.

See Also

Viewing File History
Project Reporting

Setting the Versions Directory

Set or change the path to the versions directory for a project. If you are changing an existing project path, the directory display shows the current path.

Dialog Box Options

Directories
Displays the current path as you navigate the directory structure below.

Directory list
Displays your system's directory structure. Double-click on drive and directory names to build the path in the text box above.

New Directory
Allows you to create a new directory without exiting from the dialog. To create a new directory:

1. Navigate the directory structure above until you reach the place where the new directory will be located.

2. Type the new directory name in the text box.

3. Click the Create button.

The Create button is disabled if the New Directory text box :

- is empty
- contains an invalid directory name
- contains a directory name that already exists in the current directory

See Also
Creating Projects

Milestone...

Opens the Milestone dialog, which allows you to create and store a permanent version of all project files once they reach a state of completion.

See Also

Milestone

Reporting...

Opens the Project Reporting dialog, which allows you to specify the attributes for, and generate a report of, all project files.

See Also

Project Reporting

Creating Projects

Create a New Project
You can create a new VERSIONS/VB project from any Visual Basic project. Follow the steps below to create a VERSIONS/VB project.

To create a new Versions/VB project:

1. Launch Versions/VB.

2. Open an existing Visual Basic project.

   Or,

   a) Create a new Visual Basic project.

   b) Select **File => Save Project** from the Visual Basic menu bar and save the Visual Basic project.

3. Select **Project => Versions Directory...** from the Project window menu bar. The Versions Directory dialog opens.

4. Navigate the displayed directory structure until the static text display indicates the directory in which you want to store your version files, then click OK.

   If the destination directory does not exist:

   a) Navigate the directory structure until the static text display indicates the location where the new directory will be located.

   b) Move the insertion point to the New Directory text box and type the new directory name (follow standard MS-DOS naming conventions).

   c) To create the new directory and close the dialog, click OK.

Once you assign the versions directory to the project, all Versions/VB functions become available for that project.

Joining an Existing Project

Others in your organization may have Versions/VB projects defined. You can join these projects by accessing the .mak file stored in the versions directory for that project. Perform the following steps to join an existing Versions/VB project.

1. Launch Versions/VB.

2. Select **Project => Join...** from the Project window menu bar and browse the displayed directory structure until you locate the .mak file for the project you want to join, then click OK.

   Versions/VB displays the Working Directory dialog.

3. To specify the working directory for the project (the local directory in which you'll store your working files), navigate the displayed directory structure until the static text display indicates the directory in which you want to store your version files, then click OK.

If the destination directory does not exist:

a) Navigate the directory structure until the static text display indicates the location where the new directory will be located.

b) Move the insertion point to the Make New Directory text box and type the new directory name (follow standard MS-DOS naming conventions).

c) Click Create to create the new working directory, then click OK.

Note To avoid potential problems with files that have been added to Visual Basic projects, it is important that all team members create parallel directory structures for shared projects.

When you click OK, Versions/VB checks out the project .mak file into your working directory, scans the file, and automatically checks out the project files.

Note Files that have been added to the Visual Basic project (using the Visual Basic **File => Add File...** option) are referenced by absolute path in the project .mak file. Versions/VB recreates the added file's directory structure and stores the checked out version using the same path information.

Sent Note to...

Opens a connection to Microsoft Mail, which allows you to prepare and send a mail message.

Purging Files

Removes all file versions from the versions directory except permanent versions and the most recent version.

Dialog Box Options

Project
Displays the current project name and path.

Files in Versions Directory & Selected Files
Project files contained in the versions directory appear in the list box on the left. Enter files to be purged into the list box on the right.
➪Copies the highlighted file from the Versions Directory list box to the Selected Files list box.
➪Copies all files from the Versions Directory list box to the Selected Files list box.
➪Clears all files from the Selected Files list box.
➪Clears the highlighted file from the Selected Files list box.

See Also
Why Purge Files?

Versions Menu ➪

Selections on the Versions menu are:

- Check-In Wizard...
- Check-Out Wizard...

- Maximum Versions...
- Purge...
- History...
- Locks...
- Permanent...
- Description...
- Delete

- Diff (Text Files Only)...

Project Options...

Allows you to specify whether to include .vbx and/or .dll files as candidates for version control.

See Also

Setting Project Options

Options...

Opens the Options dialog, which allows you to configure VERSIONS/VB to suit your personal needs.

Refresh

Instructs VERSIONS/VB to rescan the project .mak file to reflect any changes to file status in the VERSIONS/VB "book" display.

See Also
Visual File Attributes (Books)

Commonly Asked Questions and Answers

Question: **When I check out an older version of a form or module I don't always see the correct version when I open it. Is the correct version being checked out?**

Answer: Yes. Visual Basic caches forms and modules if enough memory is available. This can lead to a minor problem if you check out a version of a form or module that is different from the one cached in memory. To refresh the cache, simply reopen the project using the Visual Basic **File => Open Project...** menu item.

Question: **How does VERSIONS/VB determine if I'm using Microsoft Mail for Windows?**

Answer: VERSIONS/VB checks to see if you have the file, MAPI.DLL, in your Windows directory. If it's there, VERSIONS/VB will log into mail. If not, all VERSIONS/VB mail functions are disabled.

Question: **When I am entering file descriptions through the Description dialog, I would like the dialog to remain open so I can enter a description for more than one file. Can this be done?**

Answer: Yes. By default, VERSIONS/VB always closes this type of dialog box. However, you can control this through the Options dialog by enabling Sticky Dialogs. This causes the Description, Check in, Check out, and other similar dialogs to remain active after the operation is complete.

Question: **For most of my files, I want to keep just three versions. However, for a few, I would like to keep six versions. Is there any way to do this?**

Answer: Yes.

When you define a new project:

1. From the Options dialog, set Default maximum versions to 3.
2. Check in all your project files.
3. Activate the Maximum Versions dialog box, select a file and enter 6 into the Maximum # of versions text box.
4. Click OK.

Question: **How many versions of a binary file should I keep?**

Answer: This is up to you. Remember, VERSIONS/VB only keeps the versions you check in. For binary files, we recommend keeping at least 5 versions and not more than 20 versions as a general rule. Be aware that you may, at any time, designate versions as permanent.

These versions are always retained regardless of the number of versions you keep.

Question: **How many versions of an text (ASCII) file should I keep?**

Answer: This is up to you. Remember, VERSIONS/VB only keeps the versions you check in. For text files, we recommend keeping **all** (zero) versions as a general rule. Be aware that you may, at any time, designate versions as permanent. These versions are always retained regardless of the number of versions you keep.

Question: **How often should I check in my files?**

Answer: Files should only be checked in when they are at a specific point of completion. Remember that when you have reached the maximum number of versions, the oldest version is removed to make room for the new version. Therefore, you do not want to check in files every time you make minor changes.

Question: **I have just completed my software project. All my latest files are checked in. I want to keep this set of important files around forever. How do I do this?**

Answer: To create an milestone, and optional archive, of your current project, select **Project ⇨ Milestone...** from the Project window menu bar.

Question: **I realize that creating a milestone for a project takes a snapshot of all of the files in the project and creates a permanent version for each file. How will I be able to remove that permanent version later and still keep all of my temporary versions?**

Answer: You can delete the milestone version through the Delete version dialog by selecting it from the Version Label drop-down list box. **Use extreme caution** when deleting permanent versions. Remember, they were made permanent for a reason.

Question: **I keep checking out files, then making changes only to find out someone else has the file locked. How can I avoid this?**

Answer: We recommend that you lock files before you begin making changes. VERSIONS/VB will then notify other users that the file is locked by you.

If the file is locked by another user, do not begin making changes to it until the other user releases it. The VERSIONS/VB History report displays the name of the user who currently has the file locked. Ask that team member to release the file before you begin to make changes.

Question: **Does everyone have to have the same working directory for the project?**

Answer: No. However, because Visual Basic allows you to include files from other projects using the **File ⇨ Add File...** option, and those files are referenced in the project .MAK file by absolute path, we strongly recommend it.

Question: **Can VERSIONS/VB handle multiple directories?**
Answer: Yes. A VERSIONS/VB project can have multiple working directories because all files referenced in the project .mak file, including those files added to the project using the Visual Basic **File ⇨ Add File...** option, are considered to be part of the VERSIONS/VB project. however, VERSIONS/VB stores all of the versions files for a project in a single versions directory.

Question: **How can I tell who has a file locked?**

Answer: Select the file and press the History button in the VERSIONS/VB Project window toolbar.

Question: **How can I print the list of all project files with descriptions attached?**
Answer: Access the Project Reporting dialog, check Description, and click OK. You can print the report by clicking the Print ⇨ button in the VERSIONS/VB Edit window.

Question: **Where does VERSIONS/VB store user options and project information?**

Answer: VERSIONS/VB stores user options and project specific information in a file called versvb.ini which is always stored in your Windows directory.

Question: **When I try to access a versions directory file (i.e., check-in, check-out, lock, etc.) I get a message telling me the file is in use by another user. I know I'm the only one using this file. How do I get rid of this message?**

Answer: While VERSIONS/VB is accessing a Versions file it locks this file using a semaphore lock file. This semaphore lock file is named based on the following rules:

filename._	If file has no extension or a one letter extension
filename.x_	If file has a two letter extension, such as .xl
filename.do_	If file has a three letter extension, such as .doc

If the machine crashed while VERSIONS/VB was checking a file in, locking it, etc., this semaphore lock file remains. You must delete it using the Windows File Manager or from the MS-DOS prompt. Do not delete this file until you are sure that no other user has a lock on the versions directory file.

Check Out

Performs a quick check-out of the selected file.

> **Note** Quick check-out is a shorthand method that is less flexible than using the Check-out Wizard ➪. You can also use the toolbar button to check out a single file.

Send...

Opens a connection to Microsoft Mail, which allows you to attach the currently selected file to a mail message.

Check-Out Wizard...

Opens the Check Out Wizard dialog so you can copy previously stored file versions into your working directory.
➪

This is the same as clicking the Check-Out Wizard button on the VERSIONS/VB toolbar.

See Also
Checking Out Files

Locks...

Opens the Locks dialog, which allows you to lock or unlock one or more files.

See Also

Locks
Why Lock Files?

Running VERSIONS/VB with Windows NT 3.1

VERSIONS/VB 1.1 runs on Microsoft Windows NT 3.1 as a 16-bit application in a Windows 3.1 subsystem. The following inconsistencies have been detected when you run VERSIONS/VB under the Windows NT 3.1 operating system due to idiosyncrasies in the Windows 3.1 operating subsystem:

- After comparing two versions of a file, VERSIONS/VB does not always remain the topmost window. If this occurs, click on the VERSIONS/VB Project window or use the Alt-Tab key to switch back to VERSIONS/VB

- Compression of a Project Milestone archive is disabled under Windows NT due to problems with tracking task handles of 16 bit MS-DOS applications (e.g., COMPRESS.EXE) launched from VERSIONS/VB. If you want your Project Milestone archive compressed under NT, perform the compression manually using whatever, compression utility you prefer.

Version Control Environment ➪

VERSIONS/VB dialogs

- Check In Wizard
- Check Out Wizard
- Delete Version
- Descriptions
- Diff
- History Report
- Locks
- Maximum # of Versions
- Milestone
- Project Options
- Project Report
- Project Window
- Purge
- User Options

VERSIONS/VB menus

- Versions
- Project
- User

View Form

Opens the Visual Basic Form window for the currently selected file.

View Code

Opens the Visual Basic Code window for the currently selected file.

Diff Examples ➪

- To compare the current working file to the latest checked-in version:
  a) Select the filename from the Versions Directory list box. VERSIONS/VB enters the number of the latest checked in version as the First Version.
  b) Leave Second Version blank and click OK. VERSIONS/VB starts the text comparison utility and begins comparing the files.

- To compare the current working file to an older checked-in version:
  a) Select the filename from the Versions Directory list box. VERSIONS/VB enters the number of the latest checked in version as the First Version.
  b) Open the First Version drop-down list and select the version number to compare.
  c) Leave Second Version blank and click OK. VERSIONS/VB starts the text comparison utility and begins comparing the files.

- To compare two previously checked-in versions of a file:
  a) Select the filename from the Versions Directory list box. VERSIONS/VB enters the number of the latest checked in version as the First Version.
  b) Open the First Version drop-down list and select a version number to compare.
  c) Open the Second Version drop-down list and select another version number to compare.
  d) Click OK. VERSIONS/VB starts the text comparison utility and begins comparing the files.