

FastWave
Version 1.0
By Bob Dolan

[Overview](#)

[How to use the system](#)

[Function Reference](#)

[How to Register](#)

Overview

Thank you for trying [FastWave](#). This is a DLL that provides easy management of WAV files for an application, as well as improved playing performance. The basic principle behind these features is in the fact that [FastWave](#) uses collections of WAV files known as Vaults. A Vault is simply a group of WAV files bundled into one file having a VLT extension. Included in the [FastWave](#) package is a utility called the Vault Builder. This application helps you to manage these Vault files. It supports creating Vaults, and allows you to add, remove, play and rename WAV files. The issue of renaming the sounds touches on one of the other benefits of the [FastWave](#) system. When a WAV file is added to a Vault, you give it a name. This name can be any description up to 20 characters. This is a lot better than the eight character limit that a filename imposes. You will have the flexibility to give your sounds meaningful names, allowing you to organize them better.

When a Vault file is used in an application, you only need to carry around one file to support your sound effects, rather than having dozens of WAV files in your installation directory. When your application loads a Vault file, all of the sounds are in memory. If you then use [FastWave](#) to play a sound, it is played from memory. There is no disk access involved, so you get instant sound response, regardless of the capabilities of your customer's disk system.

So, if you are developing an application that uses sound, you can see the advantage that the [FastWave](#) system can provide. To summarize the system's benefits:

- Instant Sound
- 20 character descriptive sound names
- All WAV files bundled into one
- An easy to use sound interface
- Vault Builder to organize your sounds

Good Luck! and have fun!

How to use the system

FastWave is a Dynamic Linked Library (a DLL). It has functions built into it that your application can call. There are two sets of functions. One for Visual Basic and one for all other languages. Both sets have the same basic function names, except that those in the Visual Basic set are preceded with a VB.

A library reference follows, but most programmers would agree that the best way to learn how to use a new tool is to see examples of usage. For this reason, the VB source for the Vault Builder application is included in the shareware package. In addition, the C source for a smaller test application is also included. You may use and modify this code to suit your needs.

Adding a DLL to a C project is beyond the scope of this manual. For Visual Basic, all you need to do is include the FASTWAVE.BAS file to your project and you can start making function calls into the **FastWave** library. In both cases, you must Load a Vault before you can Play any of it's sounds. In addition, you must issue a StopFastWave and an UnLoadFastWave before your application exits, so that resources allocated by **FastWave** can be freed. This is best done in you main form's Unload event (in VB) or in your WM_DESTROY message handler (in C).

Function Reference

All functions return a zero on failure, non-zero on success.

int **LoadFastWave** (char *fname);

VBLoadFastWave (ByVal fname\$) As Integer

Loads the Vault file specified by the fname parameter. This function allocates memory. Before your application terminates, be sure to free this memory by calling UnLoadFastWave or VBUnLoadFastWave.

int **CreateFastWave** (char *fname);

VBCreateFastWave (ByVal fname\$) As Integer

Creates an empty Vault file with the name and at the path specified by the fname parameter. This function unloads any loaded Vault file. To access this new vault, you must load it first.

int **AddFastWave** (char *fname, char *FWname, char *wname);

VBAddFastWave (ByVal fname\$, ByVal FWname\$, ByVal wname\$) As Integer

Adds a sound (the WAV file specified by the fname parameter) at the first available slot in the Vault file specified by the FWname parameter, and gives it the description specified in the wname parameter. This function unloads any loaded Vault file.

int **ReplaceFastWave** (char *fname, char *FWname, char *wname, int wn);

VBReplaceFastWave (ByVal fname\$, ByVal FWname\$, ByVal wname\$, ByVal wn As Integer) As Integer

Replaces the sound number specified by the wn parameter with the WAV file specified by the fname parameter in the Vault file specified by the FWname parameter, and gives it the description specified in the wname parameter. This function unloads any loaded Vault file.

int **DeleteFastWave** (char *FWname, int wnum);

VBDeleteFastWave (ByVal FWname\$, ByVal wnum%) As Integer

Deletes the sound number specified by the wn parameter in the Vault file specified by the FWname parameter.

int **PlayFastWave** (int wnum, int mode);

VBPlayFastWave (ByVal wnum%, ByVal mode%) As Integer

Plays the sound number specified by the wnum parameter of the currently loaded Vault file. The mode type is one if the following values:

- 1-SYNC: The sound plays to completion, and cannot be interrupted.
- 2-ASYNC: The sound plays to completion, but can be interrupted by a subsequent Play command.
- 3-LOOP: The sound plays to completion and then plays again. This continues until another Play command is issued or a Stop command is made.

int **StopFastWave** (void);

VBStopFastWave () As Integer

Stops the currently playing sound.

int **UnLoadFastWave** (void);

VBUnLoadFastWave () As Integer

Removes the currently loaded Vault file and deallocates any memory that it used.

int **FastWaveInfo** (char *name, int wn);

VBFastWaveInfo (ByVal fname\$, ByVal wn%) As Integer

Returns the description of the sound number specified by the wn parameter of the currently

loaded Vault file. The description is placed in the name parameter.

int **DescFastWave** (char *name, int wn);

VBDescFastWave (ByVal fname\$, ByVal wn%) As Integer

Changes the description of the sound number specified by the wn parameter of the currently loaded Vault file with a new description as specified by the name parameter.

int **ExportFastWave** (char *fname, int wn);

VBExportFastWave (ByVal fname\$, ByVal wn%) As Integer

Exports the sound number as specified by the wn parameter of the currently loaded Vault file to a WAV file as specified by the fname parameter.

Registration Form

To print this registration form, open the **File** menu and select **Print Topic**.

Mail to:

Bob Dolan
P. O. Box 16514
Rochester, New York 14616-0514

Please register me as a satisfied user of [FastWave](#) (ver 1.0):

Amount Enclosed:

\$25 [] Registration includes the latest version of FastWave and a Vault file containing over 50 sounds that are perfect for most applications. In addition, the copyright dialog is not shown. Registered users will get my undivided attention for support and requested customization, although I will try to help anyone who truly intends to register.

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Phone(optional): _____

AOL/GEnie/CompuServe/Internet mail address: _____

Where did you get [FastWave](#)? This is important in that it helps us to distribute the product to places where you are more likely to see it.

Comments: _____

Registering FastWave

FastWave is copyright 1995 by Bob Dolan

FastWave is distributed as shareware. This means that if this library is used in any commercial software or Shareware, **then you must register by sending \$25 to the author.** Make checks payable to Bob Dolan and send them to the address shown on the registration form.

Please print the [registration form](#) and fill in the requested information

You may reach me on at the following EMail addresses:

GEnie: DOLAN

Compuserve: 71075,3256

Internet: bdolan@rochgte.fidonet.org

Distribution

If you distribute FastWave to friends, associates, or to a computer bulletin board system (BBS), you may only distribute the compressed file including the copyright and disclaimer.

Disclaimer

The Copyright Owner hereby disclaims all warranties relating to this software, whether express or implied, including without limitation any implied warranties of merchantability or fitness for a particular purpose. The Copyright Owner will not be liable for any special, incidental, consequential, indirect or similar damages due to loss of data or any other reason, even if The Copyright owner or an agent of The Copyright Owner has been advised of the possibility of such damages. In no event shall The Copyright Owner's liability for any damages ever exceed the price paid for the license to use the software, regardless of the form of the claim. The person using the software bears all risk as to the quality and performance of the software.

