WavPlus2.DLL Help

Another Shareware Product from Digital PowerTOOLS

Overview of WavPlus2.DLL

The VB Pro version comes with the Multimedia MCI Control that allows you to play and record WAV and MIDI files. Until now, users of the VB standard version have not had this capability without resorting to the complex functionality of the Windows 3.1 MCI (Media Control Interface) API. WavPlus2 simplifies programming Wav (and Mid) playback and recording by providing an easy-to-understand command structure, complete with error-response messages. You can even playback sections of Wav files which conserves valuable disk space when you combine several waveforms into a single file. WavPlus2.DLL utilizes both MCI and low-level access. You can also easily determine if the user's system can play WAV or MID files. Includes extensive source-code demos, CD audio capability, and dozens of functions.

The demo program (written in VB 3.0 standard) demonstrates nearly all of the new Wav features of WavPlus2 and reveals a few VB programming tricks too. And, since WavPlus2.DLL comes with a complete Help file and a practical demonstration project in VB, you'll quickly understand how to use the new functions so that you can incorporate these SoundCard techniques into your own projects.

CD Audio Functions

Declare Sub CDstatusTracks Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the number of tracks on the CD
* on data (non-musical) CD's this will return "1"
* before calling, set ReturnString$=space$(255)
* after calling, use val(trim$(TrimAtNull(ReturnString$))) to obtain track count

Declare Sub CDstatusMode Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the current status of the CD audio device:
  not ready, open, paused, playing, seeking, or stopped
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) to obtain mode

Declare Sub CDstatusCurrentTrack Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the number of the current track on the CD
* before calling, set ReturnString$=space$(255)
* after calling, use val(trim$(TrimAtNull(ReturnString$))) to obtain current track

Declare Sub CDstatusTrackLength Lib "WavPlus2.DLL" (ByVal ReturnString$, ByVal TrackNbr%)

* returns the length (mm:ss:ms) specified track on the CD
* TrackNbr is the track number
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) to obtain track's length

Declare Sub CDstatusInserted Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the word "true" if CD is in drive, else it returns "false"
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) to obtain answer

Declare Sub CDstatusPosition Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the current position (track:mm:ss:ms) of the CD
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) to obtain answer

Declare Sub CDpause Lib "WavPlus2.DLL" (ByVal ReturnString$)

* pauses a CD audio session (with some drivers this is the same as stop)
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) will contain any possible error

Declare Sub CDresume Lib "WavPlus2.DLL" (ByVal ReturnString$)

* resumes a CD audio session (with some drivers this has no effect)
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) will contain any possible error

Declare Sub CDstop Lib "WavPlus2.DLL" (ByVal ReturnString$)

* stops a CD audio session
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) will contain any possible error

Declare Sub CDplay Lib "WavPlus2.DLL" (ByVal ReturnString$, ByVal Track%)

* begins a CD audio session
* Track% is the track to play (0 = all tracks)
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) will contain any possible error

Declare Sub CDgoto Lib "WavPlus2.DLL" (ByVal ReturnString$, ByVal Track%)

* moves to the specified track on the CD
* Track% is the track to relocate to
* before calling, set ReturnString$=space$(255)
* after calling, use trim$(TrimAtNull(ReturnString$)) will contain any possible error

Copyright Notice

WavPlus2.DLL, Copyright © 1994, 1995
by Digital PowerTOOLS

WavPlus2.DLL is distributed under the shareware concept. Shareware Registration is mandatory for continued use of this product.

And, as shareware, you may evaluate WavPlus2.DLL for a period of no more than 30 days. After this time you must either register it or remove it from your system. Failure to comply with this condition is a violation of United States and international copyright law.

This program is fully functional; however, you will receive the latest version upon registering (with the startup registration reminder removed). Please register if you use this program for more than the 30-day trial period. See Support Documents for more specific details on registering this product.

Wav Functions (non-MCI)

Declare Function PlayWavWait% Lib "WavPlus2.DLL" (ByVal FullPath$)

* this function plays a Wav file waiting for completion before returning control to the calling program
* FullPath$ is the full path of the Wav file to play
* returns TRUE (-1) is the Wav file can be played; returns FALSE (0) otherwise

Declare Function PlayWavNoWait% Lib "WavPlus2.DLL" (ByVal FullPath$)

* this function plays a Wav file and returns control to the calling program as soon as the Wav file begins
* FullPath$ is the full path of the Wav file to play
* returns TRUE (-1) is the Wav file can be played; returns FALSE (0) otherwise

Declare Function PlayWavLoop% Lib "WavPlus2.DLL" (ByVal FullPath$)

* this function plays a Wav file in a repeat loop; returns control to the calling program as soon as the Wav file begins
* FullPath$ is the full path of the Wav file to play
* returns TRUE (-1) is the Wav file can be played; returns FALSE (0) otherwise

Declare Function StopWavLoop% Lib "WavPlus2.DLL" ()

* this function stops a Wav file loop started with PlayWavLoop
* returns TRUE (-1) is the Wav file can be played; returns FALSE (0) otherwise

Declare Function HowManyWavPlayDevices% Lib "WavPlus2.DLL" ()

* this function determines the number of system devices that can PLAY Wav files
* returns the number: 0 for none, 1 for one, etc.

Declare Function HowManyWavRecordDevices% Lib "WavPlus2.DLL" ()

* this function determines the number of system devices that can RECORD Wav files
* returns the number: 0 for none, 1 for one, etc.

Assorted Functions

Declare Function GetWavPlusVersion% Lib "WavPlus2.DLL" ()

- * this function returns the version of WavPlus.
- * the returned value is 100 times the actual version number.
- * for instance, a returned value of 200 represents version 2.00.

Declare Sub RevString Lib "WavPlus2.DLL" (ByVal ReturnString$)

- * this function reverses a string
- * it is most useful in conjunction with VB's INSTR command; this way you can find the position of a substring searching from the end instead of the beginning
- * ReturnString$ is the string to reverse
- * using RevString twice on the same string will restore it to the original character sequence

Wav System Info Functions (MCI)

Declare Sub WavManufacturer Lib "WavPlus2.DLL" (ByVal ReturnString$)

* determines the manufacturer of the sound card ("invalid" on error)
* before using, set ReturnString=Space$(255)

Declare Function WavVersion% Lib "WavPlus2.DLL" ()

* returns the Sound Driver version (0=invalid)
* version is minor*100+major

Declare Function WavVolumeSupported% Lib "WavPlus2.DLL" ()

* returns -1 (TRUE) if volume control support; return 0 (FALSE) otherwise

Declare Sub WavCanPlay Lib "WavPlus2.DLL" (ByVal ReturnString$)

* determines if the system can play Wav files
* before using, set ReturnString$=Space$(255)
* returns "true" if the system can play Wav files; returns "false" otherwise

Declare Sub WavCanRecord Lib "WavPlus2.DLL" (ByVal ReturnString$)

* determines if the system can record Wav files
* before using, set ReturnString$=Space$(255)
* returns "true" if the system can record Wav files; returns "false" otherwise

Declare Sub WavMaxBitSize Lib "WavPlus2.DLL" (ByVal ReturnString$)

* determines the maximum bit size capability of the system
* before using, set ReturnString$=Space$(255)
* returns "8" or "16"

Declare Sub WavMaxChannels Lib "WavPlus2.DLL" (ByVal ReturnString$)

* this command determines the maximum number of channels the system is capable of
* before using, set ReturnString$=Space$(255)
* returns "Mono" or "Stereo"

Declare Sub WavMaxSampleRate Lib "WavPlus2.DLL" (ByVal ReturnString$)

* this command determines the maximum sample rate (Hz) capability for the system
* before using, set ReturnString$=Space$(255)
* returns "11025", "22050", or "44100"

Wav File Info Functions (MCI)

Only use these commands on files that are NOT open.

Declare Sub WavGetLengthMS Lib "WavPlus2.DLL" (ByVal lpFileName$, ByVal ReturnString$)

* this command obtains the length (in milliseconds) of a Wav file
* FullPath$ is the full path of the Wav file
* before using, set ReturnString$=Space$(255)
* upon completion, ReturnString$ contains the length (in milliseconds)

Declare Sub WavGetLengthBytes Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* this command obtains the length (in bytes) of a Wav file
* FullPath$ is the full path of the Wav file
* before using, set ReturnString$=Space$(255)
* upon completion, ReturnString$ contains the length (in bytes)

Declare Sub WavGetBitSize Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* this command obtains the bit size of a Wav file
* FullPath$ is the full path of the Wav file
* before using, set ReturnString$=Space$(255)
* upon completion, ReturnString$ contains the bit size; typically "8" or "16"

Declare Sub WavGetChannels Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* this command obtains the number of channels for a Wav file
* FullPath$ is the full path of the Wav file
* before using, set ReturnString$=Space$(255)
* upon completion, ReturnString$ contains the number of channels; typically "1" for Mono or "2" for Stereo

Declare Sub WavGetSampleRate Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* this command obtains the sample rate (in Hz) for a Wav file
* FullPath$ is the full path of the Wav file
* before using, set ReturnString$=Space$(255)
* upon completion, ReturnString$ contains the sample rate; typically "11025", "22050", or "44100"

Wav Open Session Functions (MCI)

Before using the Open Session commands, a Wav file MUST first be opened with WavOpen.

Before using these commands, be SURE to set ReturnString$=Space$(255). This provides an information (or error) return buffer.

Declare Sub WavOpen Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* opens a Wav file for use by the other MCI open session commands
* FullPath$ is the full path of the Wav file
* upon return, if ReturnString$ contains only CHR$(0), the file was opened successfully; otherwise ReturnString$ will contain an error message

Declare Sub WavClose Lib "WavPlus2.DLL" (ByVal ReturnString$)

* closes an opened Wav file
* upon return, if ReturnString$ contains only CHR$(0), the file was opened successfully; otherwise ReturnString$ will contain an error message

Declare Sub WavPause Lib "WavPlus2.DLL" (ByVal ReturnString$)

* pauses an open Wav file in play
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavStop Lib "WavPlus2.DLL" (ByVal ReturnString$)

* stops an open Wav file in play, and returns the position pointer to "0" milliseconds
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavStart Lib "WavPlus2.DLL" (ByVal ReturnString$)

* starts playing an open Wav file (does not wait for completion)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavStartSectionWait Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)

* plays the specified section of an open Wav file (waits for completion)
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavStartSectionNoWait Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)

* starts playing the specified section of an open Wav file (does not wait for completion)
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavSeekEnd Lib "WavPlus2.DLL" (ByVal ReturnString$)

* moves the position pointer to the end of the file
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavSeekStart Lib "WavPlus2.DLL" (ByVal ReturnString$)

* moves the position pointer to the start of the file
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavSeekPosition Lib "WavPlus2.DLL" (ByVal lpStrValue$, ByVal ReturnString$)

* moves the position pointer to the specified location (in milliseconds) in the file
* lpStrValue$ is the new location (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub WavStatusMode Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the status of the currently open Wav file
* upon return, ReturnString$ contains the current status (one of the following):
  not ready, paused, playing, stopped, recording, or seeking
* unless you record and playback Wav files in a self-contained form such as the VB demo program uses; you should test the current status in a Timer event. MCI commands can

not open a new file until the last one is closed.

Declare Sub WavStatusLengthMS Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the length (in milliseconds) of the open Wav file

Declare Sub WavStatusLengthBytes Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the length (in bytes) of the open Wav file
* note: this is not the file size; it is the size of the recorded Wav data

Declare Sub WavStatusPosition Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the current position of the file (in milliseconds)

Declare Sub WavStatusBitSize Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the bit size of the file; typically "8" or "16"

Declare Sub WavStatusChannels Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the channel size of the file; typically "1" for Mono or "2" for Stereo

Declare Sub WavStatusSampleRate Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the sample rate of the file (in Hz); typically "11025", "22050", or "44100"

Declare Sub WavRecord Lib "WavPlus2.DLL" (ByVal ReturnString$)

* begins recording at the current file position
* note: the recording is not permanent until saved

Declare Sub WavSave Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* stores the Wav file to disk
* FullPath$ is the full path of the Wav file

Declare Sub WavEraseAll Lib "WavPlus2.DLL" (ByVal ReturnString$)

* erases the entirety of the current Wav file
* note: this is not permanent until saved

Declare Sub WavEraseSection Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)

* erases a section of the current Wav file
* note: this is not permanent until saved
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)

Declare Sub WavSetBitSize Lib "WavPlus2.DLL" (ByVal lpValue$, ByVal ReturnString$)

* sets the bit size for a new file
* lpValue$ will either be "8" or "16"; check lpString$ for an error message

Declare Sub WavSetChannels Lib "WavPlus2.DLL" (ByVal lpValue$, ByVal ReturnString$)

* sets the number of channels for a new file
* lpValue$ will either be "1" (for Mono) or "2" (for Stereo); check lpString$ for an error message

Midi Info Functions (non-MCI)

Declare Function HowManyMidiPlayDevices% Lib "WavPlus2.DLL" ()

* this function determines the number of system devices that can PLAY Mid files
* returns the number: 0 for none, 1 for one, etc.

Declare Function HowManyMidiRecordDevices% Lib "WavPlus2.DLL" ()

* this function determines the number of system devices that can RECORD Mid files
* returns the number: 0 for none, 1 for one, etc.

Midi Open Session Functions (MCI)

Before using the Open Session commands, a Mid file MUST first be opened with MidiOpen.

Before using these commands, be SURE to set ReturnString$=Space$(255). This provides an information (or error) return buffer.

Declare Sub MidiOpen Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)

* opens a Mid file for use by the other MCI open session commands
* FullPath$ is the full path of the Mid file
* upon return, if ReturnString$ contains only CHR$(0), the file was opened successfully; otherwise ReturnString$ will contain an error message

Declare Sub MidiClose Lib "WavPlus2.DLL" (ByVal ReturnString$)

* closes an open Mid file
* upon return, if ReturnString$ contains only CHR$(0), the file was opened successfully; otherwise ReturnString$ will contain an error message

Declare Sub MidiPause Lib "WavPlus2.DLL" (ByVal ReturnString$)

* pauses a playing Mid file

Declare Sub MidiStop Lib "WavPlus2.DLL" (ByVal ReturnString$)

* stops a playing Mid file and return to position pointer to "0"

Declare Sub MidiStart Lib "WavPlus2.DLL" (ByVal ReturnString$)

* starts playing a Mid file (does not wait for completion)

Declare Sub MidiStartSectionWait Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)

* plays the specified section of an open Mid file (waits for completion)
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub MidiStartSectionNoWait Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)

* starts playing the specified section of an open Mid file (does not wait for completion)
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub MidiSeekEnd Lib "WavPlus2.DLL" (ByVal ReturnString$)

* moves the position pointer to the end of the open Mid file

Declare Sub MidiSeekStart Lib "WavPlus2.DLL" (ByVal ReturnString$)

* moves the position pointer to the start of the open Mid file

Declare Sub MidiSeekPosition Lib "WavPlus2.DLL" (ByVal lpStrValue$, ByVal ReturnString$)

* moves the position pointer to the specified location (in milliseconds) in the file
* lpStrValue$ is the new location (in milliseconds)
* upon return, ReturnString$ will contain an error message (if applicable)

Declare Sub MidiStatusMode Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the status of the currently open Mid file
* upon return, ReturnString$ contains the current status (one of the following):
not ready, paused, playing, stopped, recording, or seeking
* unless you record and playback Mid files in a self-contained form such as the VB demo program uses; you should test the current status in a Timer event.   MCI commands can not open a new file until the last one is closed.

Declare Sub MidiStatusLengthMS Lib "WavPlus2.DLL" (ByVal ReturnString$)

* returns the length of the open Mid file (in milliseconds)

```
Declare Sub MidiStatusPosition Lib "WavPlus2.DLL" (ByVal ReturnString$)
```

* returns the current position of the open Mid file (in milliseconds)

```
Declare Sub MidiRecord Lib "WavPlus2.DLL" (ByVal ReturnString$)
```

* begins recording at the current file position of the Mid file

```
Declare Sub MidiSave Lib "WavPlus2.DLL" (ByVal FullPath$, ByVal ReturnString$)
```

* stores the current Mid file
* FullPath$ is the full path of the Mid file

```
Declare Sub MidiEraseAll Lib "WavPlus2.DLL" (ByVal ReturnString$)
```

* erases the entirety of the currently open Mid file

```
Declare Sub MidiEraseSection Lib "WavPlus2.DLL" (ByVal lpSecStart$, ByVal lpSecEnd$, ByVal ReturnString$)
```

* erases a section of the current Mid file
* lpSecStart$ is the starting position (in milliseconds)
* lpSecEnd$ is the ending position (in milliseconds)

Support Documents

Included with the WavPlus2.DLL demo are five support files in WRI format; you can also access these files directly from the Help menu of the demonstation application.   These files are:

DpCt0395.WRI (a description of other Digital PowerTOOLS shareware products)

EvalFrm.WRI (a shareware evaluation form for you to grade this product)

OnlineRg.WRI (explains how to register online through CompuServe)

OrderFrm.WRI (a convenient order form with bonus and upgrade discounts)

SharWare.WRI (a definition of shareware and a description of our policy on shareware)