QuickTone Prototyper for Windows

Help System - Contents

_____

General Topics

Menu Items

Introduction

_____

QuickTone Prototyper for Windows is a stand-alone programming utility designed to simplify the process of adding PC speaker sound effects to applications written for the Microsoft Windows operating environment.

For many Windows programmers, the use of the standard PC speaker is limited to simple calls to the Windows API "Message Beep" function. While useful, Message Beep has no provision for altering the *tone* or *duration* of the sound it produces.

The creation of more sophisticated sound effects requires an understanding of other API functions, including SetVoiceNote, SetVoiceAccent, and SetVoiceSound.

Of these, SetVoiceNote offers probably the most straightforward control over PC speaker tones. Even so, the production of a single note from the PC speaker requires the setting of several SetVoiceNote parameters, **and** its use in conjunction with the OpenSound, StartSound, WaitSoundState, StopSound, and CloseSound functions. Considerable programming time can be spent writing and compiling code over and over again to produce the desired tune or sound effect.

The QuickTone Prototyper offers a quick and easy alternative.

QuickTone allows the user to write, edit, test, and record individual series (voice queues) of notes using calls to Windows API functions. Best of all, QuickTone can write the finished lines of API functions to individual text files for later use *or* copy the lines directly into the Windows Clipboard for one-step pasting *directly into your source code text editor*.

Note: QuickTone's use of standard API functions ensures compatibility with virtually any application programming language that makes calls to the Windows API, including:

Microsoft SDK for Windows
Microsoft Quick C for Windows
Borland C++
Borland Turbo Pascal for Windows
...and many others.

The QuickTone Interface

QuickTone's on-screen interface is divided into four main areas.

The **Note Select** area is used for selecting a specific note in an active voice queue. A maximum of eight possible notes may be placed into any single voice queue, but only one note at a time may be edited. The controls in this area allow a selected note to be test-played or erased at any time.

The **Queue Control** area contains two status fields for displaying the active voice queue and the number of notes it contains. A scroll bar control is provided for manually setting the number of active notes in a given queue. In addition, three command buttons are provided, allowing the user to *display* the active queue in a dialog box window, *save* the active queue to the Windows clipboard, or *write* the active queue to an ASCII text file.

The **Queue Select** area is similar to the Note Select area. It is used for selecting a specific voice queue for editing. A total of eight voice queues are available to QuickTone. The controls in this area allow test-playing of individual voice queues or sequence-playing of all active queues. A third control button allows the user to clear the contents of all eight voice queues at once.

The **Active Note Settings** area is used for adjusting the particular sound settings of the currently active note. Three scroll bars are provided for this purpose. In addition, two status windows are used to display the selected note and its SetVoiceNote parameters during the editing process.

Using QuickTone

QuickTone Prototyper pre-selects voice queue number 1 when the program is loaded. At that time, all queues are considered empty and each note in every queue is considered vacant.

To select a note or notes to edit, simply "click" one of the numbered buttons in the "Note Select" section. The "Notes in Queue" scroll bar and status window will be automatically updated to reflect the number of active notes in the current queue.

The scroll bar controls the number of active notes assigned to the current voice queue.

Note: The Note Select control buttons will override the Notes in Queue controls if the note selected for editing is greater than the Notes in Queue setting. Both the scroll bar position and the status field will automatically reflect the addition of new notes. Conversely, the status field will automatically reflect the erasure of existing notes via the Notes in Queue scroll bar.

When a note is selected (active), the actual editing is done through the "Active Note Settings" area at the bottom of the interface screen. Here, each of the three adjustable SetVoiceNote parameters may be set. As each value is adjusted, the SVN function displayed in the status window is updated to reflect the current function parameters. Specific information on the effect of each of the three Active Note Settings controls is provided in the "SetVoiceNote Explained" section.

The active (currently selected) note may be test-played at any time by clicking the "Play Active Note" button. In addition, the entire active queue may be played at any time by clicking the "Play Active Queue" button. **All** queues containing active notes may be played in sequence by clicking the "Play All Queues" button.

Note: Voice queue notes with a "Note Value" of zero (0) will be "played" as rest notes or pauses.

To clear the active note, simply click the "Clear Active Note" button. This resets the note's SetVoiceNote parameters to the default "vacant" values.

To clear **all** notes in the active voice queue, click the "Clear All Notes" button. This resets all notes in the queue to their default values. The active queue may also be cleared by manually sliding the "Notes in Queue" scroll bar to its zero position.

QuickTone How-To...

Select a Note
Individual notes are selected for editing or playing by "clicking" any of the numbered Note Select control buttons. The status field in the Active Note Settings area will be updated to reflect your selection.

Select a Voice Queue
Queue #1 is automatically opened for editing when the program is first loaded. To select a different queue, simply "click" one of the eight Queue Select control buttons. The status field in the Queue Control area will be updated to reflect your selection.

Test-Play Individual Notes
The current active note may be played by clicking the **Play Note** control button.

Test-Play the Voice Queue
The entire voice queue may be played by clicking the **Play Queue** control button.

Test-Play All Voice Queues in Sequence
The contents of all voice queues may be played in their entirety by clicking the **Play All Queues** control button.

Display/View the Entire Voice Queue
The **Display Queue** control button displays the entire active voice queue and supporting API functions in a dialog box.

Save the Voice Queue Module to the Windows Clipboard
The **Save to Clipboard** control button copies the contents of the active queue to the Windows Clipboard for direct pasting into your Windows-compatible source code editor.

Save the Voice Queue Module to a Text File
The **Write to File** control button writes the active queue to an ASCII text file (compatible with Windows Notepad, or any other source code text editor) for archiving and/or later use.

The "Display Queue" Option

QuickTone automatically displays the API SetVoiceNote function parameters of the current active note in the Active Note Settings area. This is essential to the process of editing the individual note, but it doesn't provide a "big picture" of the entire active voice queue module.

Activating the **Display Queue** option in the Queue Control area will display *every* SetVoiceNote function in the active voice queue, as well as their "supporting" API sound functions. This collection of API code is called a QuickTone module, and its display here represents a "preview" of the text that may be written to a file or copied to the Windows clipboard through the other two Queue Control command buttons.

The functions that make up a module constitute all the code required to generate the exact sounds prototyped in QuickTone. Depending on your development language and compiler, the API code thus generated may be laid directly into your source code with little or no modification.

Note: Your programming language may require a slightly different syntax to the API sound functions than is represented here. If this is the case, the most convenient way of prototyping sound effects would be to construct them in QuickTone, save them as ASCII text, and then make the necessary modifications in a text editor.

.

The "Save to Clipboard" Option

When a desired note or notes have been entered into a voice queue, and the queue has been satisfactorily tested, the user has the option of saving the voice queue "module" to the Windows clipboard or to an ASCII text file.

If your source code text editor runs under the Microsoft Windows operating environment, the easiest way to transfer a completed module is via the Windows clipboard.

The **Save to Clipboard** option saves a text module containing all SetVoiceNote functions in the active voice queue, as well as their supporting API functions, into the Windows clipboard. The module may then be "pasted" directly into your source code text editor through the editor's menu bar "Edit/Paste" command.

Note: The Save to Clipboard option can only copy one voice queue module at a time, and will clear the clipboard of any data previously copied there. This option will **not**, however, erase any QuickTone voice queue information residing in memory; only the Clear Note or Clear Queue commands can do that.

The "Write to File" Option

QuickTone's **Write to File** option allows the user to save complete stand-alone modules of Windows API functions to an ASCII text file. The modules are comprised not only of the SetVoiceNote function(s) prepared in QuickTone, but the supporting API functions as well, and constitute all the necessary API calls required to produce the sound effects prototyped in QuickTone.

Files saved in this manner are written to the directory containing the QuickTone program files in the format: **QUEUE_*.TXT** (e.g., QUEUE_01.TXT, QUEUE_02.TXT, etc.).

If you would like to save voice queue modules for archiving purposes, the Write to File option is a useful alternative to the Save to Clipboard command.

Note: Your programming language may require a slightly different syntax to the API sound functions than is represented here. If this is the case, the most convenient way of prototyping sound effects would be to construct them in QuickTone, save them as ASCII text, and then make the necessary modifications in a text editor.

The Voice Queue

What are *Voice Queues*, and how are they used?

Individual notes, established through the SetVoiceNote function in QuickTone, are placed in what is called a "Voice Queue" at program run-time.

Because several notes may be grouped together and placed in a single voice queue for sequential playing, it might help to think of the voice queue as an "envelope" that holds a set of notes until it is time for them to be played.

Once notes have been assigned to a given voice queue, it is the responsibility of the supporting API functions (like *WaitSoundStates*) to control how the queue is played and whether its contents are maintained in memory or cleared.

SetVoiceNote Explained

The QuickTone Prototyper constructs PC speaker tone source code via the Windows API "SetVoiceNote" function. SetVoiceNote parameters set not only the tone note itself, but the duration of the tone and the "blank" (rest) just before and after the tone.

Syntax

SetVoiceNote(Voice, Value, Length, CDots);

Parameters

Voice
A specified voice queue. For our purposes, this value is preset to 1.

Value
An integer note value. When this value is set to zero, a rest ("blank") note is played. A value in the range 1-84 will produce a tone; generally higher in pitch as the value setting is increased.

Length
An integer value reciprocal of note length duration. This value exercises primary control over the duration of the played note; generally shorter as the length setting is increased. Length settings below 12 will allow a note duration of a full second or more.

CDots
An integer value of note duration in CDots. This value exercises secondary control of the duration of the played note, and may be used in conjunction with the "Length" setting to alter the rest period just before or after the note iself is played. For most purposes, however, this setting may be left at 1, with the "Length" value controlling note duration exclusively.

API Functions Used by QuickTone

_____

The QuickTone Prototyper makes use of several Windows API functions in conjunction with SetVoiceNote to prepare the voice queue to receive note information and then play the note or notes in the specified queue correctly at run-time. By themselves, none of these functions can actually produce a recognizable tone through the PC speaker. Together, however, they may be used to create a wide variety of effects.

The essential functions used in QuickTone are:

OpenSound;
This function enables the play device for use by the application.

SetVoiceNote(Voice, Value, Length, CDots);
This function establishes the parameters for a single given note (see "SetVoiceNote Explained").

StartSound;
This function begins the voice queue "play" process.

WaitSoundStates(S_QueueEmpty);
This function specifies the play driver's end-play state. The "S_QueueEmpty" parameter instructs Windows to play the contents of the voice queue once, without stopping, until every note in the queue has been played.

StopSound;
This function ends the playing of the voice queue.

CloseSound;
This function empties the voice queue and frees the used buffers, effectively restoring the program processing to the state that existed prior to sound operations.

Note: Examples of how these API functions are used together are provided in the Help "Examples" section.

Examples

_____

The following example illustrates the functions and parameters required to place a note into the voice queue and play it:

```
.
OpenSound;
SetVoiceNote(1, 20, 90, 1);
StartSound;
WaitSoundState(S_QueueEmpty);
StopSound;
CloseSound;
.
```

This example will produce a short-duration tone comprised of a single high-pitched note.

The following example illustrates the functions and parameters required to place multiple notes into the voice queue and play them in succession:

```
.
OpenSound;
SetVoiceNote(1, 20, 30, 1); { A low note }
SetVoiceNote(1, 40, 30, 1); { A medium note }
SetVoiceNote(1, 60, 30, 1); { A high note }
StartSound;
WaitSoundState(S_QueueEmpty);
StopSound;
CloseSound;
.
```

As you can see, it is fairly simple to place multiple "SetVoiceNote" notes into the voice queue. Just remember that the notes are played in the order they appear in the source code. For the most part, the other API functions remain the same whether the queue contains one note or a dozen. For our purposes, however, QuickTone will limit you to a total of eight notes per voice queue.

Notice, too, how the note "Value" settings are somewhat higher for each of the three notes. This example will produce medium-duration tones comprised of three progressively higher-pitched notes.

```
.
```

Exiting QuickTone

The user may exit QuickTone at any time by selecting the menu bar "File/Exit" command.

Text still residing in the Windows Clipboard will remain there following the closing of QuickTone. The Windows Clipboard may be cleared through the Clipboard menu bar "Edit/Delete" command.

Note: Text in the Clipboard is automatically overwritten when the "Edit/Paste" command is used within a Windows application.

Getting Help

Run-time help is available at any time by selecting the menu bar "Help/Contents" control. This control accesses the Windows Help system and loads the file **QTONE.HLP** (this file), which contains help on a number of topics related to QuickTone's operation.

A printable registration form is also provided in the help file.

Additional information is available in the text file, **QTONE.TXT**, which may be loaded and read with Windows Notepad.

Product Registration

QuickTone Prototyper is shareware, not freeware. Product registration is necessary if companies like ours are to continue providing quality software for the shareware marketplace.

Registered users receive the registered version of this product (available in 3.5" or 5.25" disk formats). In addition, registration entitles the user to advance information on upcoming Pegasus Development products (including scheduled upgrades to QuickTone) -- as well as other new titles in our expanding product line.

The Windows Help system you are using has a convenient feature for printing any topic or topics in this file. This feature may be used to print topic-screens for reference while using QuickTone -- and may also be used for printing the Registration Form included in this Help file (see "Registration Form").

To use this feature, select the desired topic with the Help Index (or browse forward until it appears on the screen). When the topic of your choice appears, click the Windows Help menu bar and select "Print Topic." Windows Help will prompt you for instructions.

Note: Complete registration information is also provided through the QuickTone menu bar "Help/Registration Information" command **and** in the enclosed text file, **QTONE.TXT**.

Limited Warranty and Remedies

Before release, our applications software is tested on a wide variety of hardware configurations, under the most rigorous test conditions possible. We therefore warrant that the software will perform in substantial compliance with the specifications set forth in this text and in the accompanying text file, provided that the software is used on the computer hardware and operating system(s) for which it was designed.

Pegasus Development makes no other warranties, expressed or implied, with respect to this software (or media, if this is a registered version), including its quality or fitness for a particular purpose.

In no event will Pegasus Development be liable for any direct, indirect, special, incidental, or consequential damages arising from the use of or the inability to use this software, even if Pegasus Development has been advised of the possibility of such damages.

Pegasus Development is not responsible for any costs including, but not limited to, those incurred as a result of lost profits or revenues, loss of time or use of the software, loss of data, the cost of recovering software or data, the cost of substitute software, claims by third parties, or similar costs. In no event will Pegasus Development's liability exceed the amount of the paid registration fee.

Registration Form

_____

Please return this form along with your remittance to:

Pegasus Development
QuickTone 1.0 Registration
11900 Grant Place
Des Peres, MO. 63131
U.S.A.

You can also register by telephone using your MasterCard or VISA through Advanced Support Group at 1-800-788-0787.

_____ QuickTone Registrations @ $16.00 U.S. each

(For orders outside the continental U.S., please add $2.00 shipping and handling)

Total enclosed . . . . . $_____

Payment by: [ ] Check [ ] MasterCard [ ] VISA

Card #_____ Exp. Date: _____

Signature of Cardholder _____

Name _____

Address _____

City _____ State _____ Zip _____

Telephone (_____) _____

Disk Size/Format Required: [ ] 3.5" (720 K) [ ] 5.25" (360 K)

Please indicate where you obtained this version of QuickTone:
(Mail Order, BBS, etc.) _____

_____

About QuickTone

QuickTone Prototyper (Version 1.0) was written in Borland's Turbo Pascal for Windows for use in the Microsoft Windows operating environment (versions 3.0 and higher).

Detailed information on the Microsoft Windows API is available from a variety of sources. We recommend the following texts:

Microsoft Windows Software Development Kit, Guide to Programming
Microsoft Corporation: 1990

Microsoft Windows Software Development Kit Reference, Vols. 1 & 2
Microsoft Corporation: 1990

For further information about this product, please contact:

Pegasus Development
Product Information
11900 Grant Place
St. Louis, MO. 63131

QuickTone Prototyper

P E G A S U S D E V E L O P M E N T