Cool Edit version 1.31

Cool Edit is a full featured wave editor for Windows, and was written by me - David Johnston.   Please read on.

Cool Features!
Common Procedures
Console Controls

File Menu
Edit Menu
Transform Menu
Generate Menu
Options Menu

Package Contents
Common Q & A
File Filters
Custom Toolbar
Updates since 0.5

Other Programs Written by Me

Note 1:
When a file is opened using COOL, a temporary file (~COL????.TMP) in your temporary directory is created.   This is where the actual wave data is going to be played from, recorded to, or edited.   This means that the largest file that COOL will work with is limited to the amount of free space on the hard drive containing the temporary directory (pointed to by the TEMP environment variable).   If you wish to save the file, you must have enough free space on the drive you are saving to.   Remember that the temporary file for COOL is taking up space on the TEMP drive, which means you cannot save a file onto your TEMP drive that is larger than one half the amount of free space you had on your TEMP drive before you started.   If COOL crashes, there may be a temp file (~COL????.TMP) in your temporary directory that you should delete.

One advantage to using this method is virtually unlimited waveform sizes, since it depends only on the size of your hard drive.   If you have a 500 meg hard drive, you can record one hour of CD quality music. The drawback, of course, is that the temporary file can get extremely large, which may not allow you to save your masterpiece on the same drive.   Of course, if you have a second hard drive with enough space, there will be no problem saving the wave.

Note 2:
If you are using a compressed, or double-spaced hard drive, there are some special considerations. Recording high-density waves (such as 44.1K/Stereo/16-bit) may severely slow down your system.   If your system appears to lock up, press the **Stop** button, then try reducing the play/record buffer size in the Settings menu.   Also be aware that the initial recording time left that is displayed when recording can be slightly off, since the amount of space audio data takes up on a compressed drive varies with the type of audio being recorded!   Cool will try and estimate the time left every few seconds, so if you see a jump in time left (either a favorable jump, or a not so favorable decrease) it is because Cool is trying the best it can to estimate the remaining time.

Note 3:
For fine tuning your system to take full advantage of both your sound card, and the COOL program, please see the Settings section.

If you wish to have COOL use a separate directory for the temporary file (other than that pointed to by the

TEMP environment string), then add the line "TempOverride = E:\newdrive" to the [Size] section in cool.ini, where E:\newdrive can be any drive, and any directory you want to use for temporary files.

Note 4:
For more accuracy in using any transformation, use 16-bit samples while working with waves.   If your board only supports 8-bit samples, you can have them converted on the fly at playback time by choosing the "convert 16 to 8 bit" mode in Settings.   To convert an 8-bit sample to 16-bit, open the waveform, copy the entire wave, then say "New" and choose 16-bit, and paste the waveform.

Warranty Disclaimer
David Johnston (*The Author*) makes no warranty of any kind, expressed or implied, including any warranties of fitness for a particular purpose.   In no event will the author be liable for any incidental or consequential damages arising from the use of, or inability to use, this program.

Cool Features

> Multiple instances (copies of Cool) can be loaded, copied to, and pasted from in any mixture of wave formats
> Cool-Edit can synthesize just about any sound using the noise and tone generation functions along with the various wave transformation functions
> Any sample can be used as an "instrument" and set to music, like a dog barking Jingle Bells, or Tarzan yelling the national anthem
> Waves of any size can be edited, limited only by hard drive space
> Supports PCM, Microsoft ADPCM and IMA/DVI ADPCM .WAV, Sound Blaster .VOC, raw PCM, and Apple AIFF data formats
> Add 1960's sounding flanging and special effects using Flange and Special Efx.
> Continuous echo of all or part of a sample with filtering for successive echoes possible with Echo.
> Speed up or slow down samples without affecting pitch, or raise and lower pitch without affecting tempo by using Stretch.
> Supports descriptive information, and bitmaps when saved in .WAV format
> Supports a cue list, and a play list when saved in .WAV format for playing portion of a wave in any order, with looping
> Built in CD player when the [MCI] CD Audio driver is loaded, and a valid music CD is in the drive
> Brainwave synchronization function available to create sound files that alter your state of mind
> Record and Playback waves using any Windows compatible sound board, even works with the PC Speaker driver
> Spatially locate sound sources to appear as if they are coming from different directions by delaying one channel a few microseconds
> Most functions have programmable Presets to save and easily retrieve your favorite effects
> You can feel free to experiment, and easily back up a step if you don't like what you did by using the Undo function
> 8-band Quick Filter for quickly adjusting equalization.
> FFT filtering and Spectral View make it easy to filter and analyze audio with high precision.
> Practically any file format can be loaded successfully into Cool by using the raw PCM file type, and then Adjusting the sample rate, bit rate, and number of channels, as well as performing Byte Swaps to get the byte ordering and interpretation correct.
> Skewed waveforms can be centered properly using the DC Bias filter, and waves can be normalized, faded in, faded out, or panned by using the Amplify function.
> Frequency analysis at the cursor can be continually be performed on the waveform at the cursor to better see exactly what frequencies are present, and where.
> Support for unlimited wave formats by allowing others to write custom file formats that Cool Edit will support.   See the File Filter API
> Cool Scripts will remember everything you did to arrive at a certain waveform, or transformation and remember it for the future.
> Customizable Toolbar (with a little work) possible by customizing the cool.ini file.

File menu

The file menu displays file handling options.

File Menu Options:
      <u>New</u>
      <u>New Instance</u>
      <u>Open</u>
      <u>Re-Open</u>
      <u>Close</u>
      <u>Save</u>
      <u>Save As</u>
      <u>Save Selection</u>
      Exit

Answers to common Questions

Q: I cannot load normal wave (*.WAV) files.   Why?
A: Do you have the file WAVE.FLT in the same directory as COOL.EXE?   The *.FLT files are required for any file loading and saving.

Q: I get an error message when I try to record.   What's up?
A: Check the capabilities of your sound card.   You may be trying to record at a rate not supported by your hardware.   Some older boards can play at higher rates than they can record.   You may also be trying to record in 16-bit mode for an 8-bit card.   When COOL is started, the playback abilities are displayed in the audio format box (but this does not necessarily mean you can record at the same rates).

Q: I get an error message when I try to play a wave.   How come?
A: Your board probably does not support the audio data sample rate, or sample size.   Try checking "Play 16-bit files as 8-bit" in the Settings box.   This will convert any 16-bit audio data to 8-bit as the file is played.

Q: I just installed a 16-bit audio card, but my 16-bit sound files still sound awful.   Should I take my card back?
A: No.   Your card is probably fine.   Check to see that the "Play 16-bit files as 8-bit" box is **not** checked in the Settings dialog.   If it is checked, your files are being converted to 8-bit before being played.   Also be sure you are using the right DMA settings.   The lower DMA channels can only support 8-bit audio.   Please check your sound board manuals for this information.

Q: Why does it take forever to do things like Filter, and to use Spectral View?
A: These functions use the Fast Fourier Transform (FFT) to convert the waveform from temporal data to frequency data.   The FFT does "zillions" of floating point operations to accomplish this.   If you do *not* have a co-processor (this includes 486SX users!) then these operations are going to take a very long time.   A co-processor will speed these operations up by *at least* a factor of 10, sometimes 20!

Q: Why does it take so long to save as ADPCM Wave format?
A: ADPCM is a compression scheme, which compresses 16-bit samples to 4-bits while retaining much of the quality at higher sample rates.   If the multiple pass option is chosen, which provides the highest quality, each block of wave data is compressed seven different ways, and the way which sounds most like the original signal is then saved.

Q: I am using Windows NT.   Why doesn't my screen redraw when I make a change?
A: Normally, COOL is set up to "multitask" when the waveform is being drawn, so as not to lock up the PC when drawing large waves.   NT automatically multitasks, which throws things out of sync.   Make sure the NT Compatibility option is checked under Settings if you experience screen redraw problems.

Q: How long did it take you to make this program?
A: It took me almost a year of programming in the evenings and weekends to produce what you see here.   I had actually started writing some of the waveform transformation functions six months prior when I wrote a DOS only version of Cool, which only worked with Sound Blaster files.   I first ported that functionality to Windows, then added on functions and features from there, and eventually ended up with this creation.   The brainwave function (Transform->Wave) was written around April to May of 1992, and eventually released to ShareWare in July of 1992.   I experimented for several months with different sound/relaxation techniques before finding the one which worked extremely well.

Q: Why are some functions not selectable?
A: If you have the unregistered version of Cool, you chose which functions you wanted to use for the editing session.   The others will not be selectable until you choose them on the next editing session.   Also, some functions only work on stereo files, such as Wave and Channel Mixer (which becomes Invert for mono waves).

Q: How can I see my wave size information in samples instead of time?
A: Double-clicking on the time (or samples Start and End) window will toggle the display between time and samples. This, and double-clicking on the wave to select all, are the only functions that do not have a corresponding menu item or button associated with them. Other shortcuts are double-clicking on the waveform type display to change the waveform interpretation (ie interpret the 44K wave as a 22K wave), and double-clicking on the green bar to bring up the viewing samples data entry box.

Q: Why do my recordings sound choppy?
A: You probably have too small of a buffer size in the Settings dialog. A minimal buffer size is about 4 seconds. Using a compressed hard drive on a slow PC could also eat up so many CPU cycles, that there isn't time left to do recording. Either try adjusting the buffer size up or down, or record at a lower data rate (ie 8 bits instead of 16, or 32K instead of 44K). I tried very hard to ensure that recordings would sound perfect, without any data loss. If you have problems, see if other recording software has the same problems. If so, you may have a hardware incompatibility between you sound card and your main board, video board, or other installed boards.

Q: My computer crashed while Cool was running, and now I lost alot of hard drive space. What happened?
A: Cool uses temporary files for the editing of waves, and the undo buffer. Since some of these files may have been open (the editing temporary buffer is always open) during the crash, the files were never closed properly. Exit Windows, and type **CHKDSK /f C:** (use whatever drive letter is missing the space). If you get "Convert lost chains to files?" or something like that, answer yes. The missing data will now be in the root directory as *.CHK files, which you can then safely delete.

Q: I keep getting "parity error" messages, and the program keeps crashing. What's going on?
A: Some older style PCs do not handle memory properly, and Cool just makes the problem worse since it is very CPU intensive. The "parity error" message appears when one of the bits that were saved into DRAM have changed unexpectedly (indicating bad memory). Most boards have the first 64K of memory on the motherboard, so the problem could be either in your memory arrays, or on the motherboard.

Q: Cool does something really weird that I didn't expect, and I think it's not working right. What's a person to do?
A: A person should send me mail explaining the bug they've found, and I will respond by either finding a fix in the program, or just letting them know that the program is supposed to do that.

Sales Pitch

I have also written a program called Cool-Edit, which includes all the functions of MindSync, plus many many more!   Some of the other features include:

- VOC, ADPCM, AIFF, and raw data wave file types
- CD Player controls
- FFT filtering, and spectral analysis
- Echo
- Flange, and other special effects
- Stretch/Squeeze and pitch change
- Harmonize, and PCM music generation
- "Hot Keys" to play wave portions
- Envelope, Reverse, and many more

If you are interested, look for Cool-Edit (*Cool???.ZIP* ?) on your favorite BBS, or drop me a line.   I will send you the unregistered version for US $5 (Cost of disk + P&H), or the registered version for US $25.

Other Programs I've Written

Keep your eyes out for other programs I've written.   If you can't find them anywhere, I will gladly send you a copy for about $5 for the disk and postage.

MindSync for Windows v.1.01
This is just like Cool Edit, except most of the fancy transformation functions have been removed.   The Wave function is the centerpiece of the program.   MindSync does not include Scripts either.   In general, Cool Edit is a superset of MindSync.   The only other difference is that I am not asking any money for the MindSync program.

MindSync for DOS v.1.01
This is the original brainwave synchronization program I wrote for DOS and the Sound Blaster.   It only generates brainwave files in 22K stereo 8-bit VOC format.   Brainwave synchronization is the same as the /i Wave function in Cool Edit.

Kaleidoscope Screen Saver v.1.40
This is a great screen saver for use with Windows.   Lots and lots of options let you customize, or completely randomize the kaleidoscopic variations.   It's hard to describe the beauty of the living, flowing artwork that *is* the Kaleidoscope screen saver.

After Dark Screen Savers
I have written three after dark screen savers: Kaleidoscope (scaled down version of the Windows counterpart), Organic (fractal designs that look like cells), and String Arts (colorful designs resembling string art -- a popular artform of the 60's and 70's where colored string is strung between nails to make beautiful patterns).

RPN Calculator
Reverse Polish Notation calculator (like HP calculators).   This calculator has dozens of great functions, including Hex/Binary/Decimal/Octal/n-ary base translations, trig functions, factorial, prime number, exponents and logs, and even biorhythms.

Wallpaper Maker
Create colorful wallpaper 'tiles' that look superb when tiled as Windows wallpaper.   Never draws the same tile twice.   See patterns in the tiles that are only visible when viewed as a group of tiles.

Classic Collection
Various DOS based graphic programs.   If you're looking for some *classic* programs that are fun to use and watch, these may interest you.   Use your mouse to control birth and death rates in a Game of Life, or place wagers on which species will win out in a different life simulation.   Steer your way through a warp speed starfield simulation.   Experiment with Iteraded Function Systems to grow plants and trees. Weave your way through the Mandelbrot set (yes, I know, there are thousands of Mandelbrot generators out there now).   Shoot marbles around the screen, and experiment with the effects of gravity.   Generate screen mazes.   Draw spiro designs, watch string art designs, or point set fractal art.   Even includes a character based 'snow' program.   Not much documentation, but most programs include the '/h' or '/help' switch for help.

Package Contents

The following files comprise the complete package for Cool Edit:

COOL.EXE	Main program
COOL.HLP	This help file
DVI.FLT	File filter to read and write DVI/IMA ADPCM *.WAV files
WAVE.FLT	File filter to read and write PCM *.WAV files
ADPCM.FLT	File filter to read and write Microsoft ADPCM *.WAV files
AIF.FLT	File filter to read and write Apple AIFF files
PCM.FLT	File filter to read and write raw PCM files
VOC.FLT	File filter to read and write standard SB Voice files
CTL3D.DLL	DLL for 3-D looking controls
*.SCP	Various script files

Updates since version 0.5

Using "Large Fonts" on some monitors correctly displays main console, as well as progress meter now.

Recording to Sound Blaster Pro works properly.   It no longer gives errors and chops up recording.   In fact, using COOL to record does a much nicer job than the recorder program provided with Sound Blaster (at least mine), and there are no "clicks" or other distractions to contend with.

Lifting up on button after dragging a selection (and leaving the main window) will now act properly, and disengage the selection process

A Recording level meter was added.   Source input can be monitored through the Options menu for setting recording levels.

A "Musical Source" checkbox was added to the brainwave transformation to eliminate clicks heard when waving musical files.

8-bit stereo pasting (with Overlap) works properly now (slight bug caused screeching problems).   Also, overlap pasting past end of file works properly too.

When saving a file, the extension .WAV is assumed if no extension is typed.

The progress meter was fixed slightly to allow multitasking with other applications without crashing Windows.

Displaying long waveforms now "multitasks" with Windows, so other things can be done while it is loading.   Also, waveform display is instantaneous once loaded in (for example, restoring a minimized long waveform displays instantly instead of recalculating).   This allows such effects as the sound level meter running while zooming in and out of a waveform.

Multiple COOL's can be opened at once (which reduced the executable as well) for working with multiple waves at once.   It's very easy to copy and paste between waves, even of different formats.

Gliding waves from one frequency to another "overshot" before.   For example, gliding from 12Hz down to 7Hz would actually glide down twice the frequency, (instead of -5Hz, -10Hz) resulting in a glide from 12Hz down to 2Hz instead.   It's fixed now.

Little Yellow Arrows work on files larger than 30,000,000 samples.

Command line loading works.   File will automatically play once loaded if another file is not already playing.   You may now associate the COOL.EXE program with .WAV files in the file manager.

Generating silence in 8-bit mode works properly now, centering the silence line at value 128.

Viewing mode of the Beginning and Ending samples, as well as the time window (which displays how long the viewed portion of the sample is, or how long the selection is) can be changed between samples and time by double clicking on the display.

The Stretch transformation was added to allow stretching and compressing of waves, adjusting pitches, or adjusting tempos.

A slight bug in overlapping was fixed.   Overlap pasting past the end of a wave that used to be longer would revive data past the end of file.

MS ADPCM Compression is supported for loading and saving compressed files now.

Extra RIFF information embedded in the .WAV file is now remembered between saves, and can be edited using the Info dialog.

The Silence   option was added to the Transform menu to quickly silence the selection portion.

The previous Save menu item was changed to Save As, and a new Save was added to save the wave currently being worked on without being asked for the file name.

Toolbar was added for quicker access to functions.

Edit keys Delete (Delete Wave Selection), Ctrl+Insert (Insert), Shift+Insert (Copy) and Shift+Delete (Paste) were added.

The RAW file format was added for saving only the wave data without any headers.

A linked list approach has been taken for working with the temporary disk file when editing.   Now Inserts and Deletes are almost instantaneous when working with very large files.(Fixed cut/paste bug from 1.01)

The Transpose function has been added to the Stretch option to musically raise or lower pitches of selected samples.

Stretching is cleaner sounding.

Music option was added to put your favorite samples to music (as if you were playing them on a keyboard) using a very easy to use music editor.   Only short riffs are supported right now.

You can use the Shift+Mouse Button to extend the highlighted selection.

Highlighting is faster for slower video cards.

Preset buttons added to Stretch dialog.

New algorithm implemented for raising/lowering pitches, and speeding up/slowing down waves.

1.03 New options in settings dialog for play/record buffer size, and highlighting after pasting.

Equalizer settings changed to decibels instead of percentages, making it easier to equalize, since the logarithmic scale is more natural.

A few new tone flavors have been added for making more natural sounding instrument sounds, which can be used with the Music option.

Envelope option was added to have more control over amplifying waves by using an amplification envelope.

The Listen option was added to the music dialog so that the notes could be played through your MIDI setup as a preview.   You can record from the music played as well (if your board supports recording from the on-board synth.)

Echo Equalizer added to Echo function to have each successive echo equalized for truer echo effects, or just snazzy special effects.

Sound Blaster VOC file compatibility was added.   Not FULL capability yet, because it does not support loops (it only loads the loop once), and ASCII text.   It *does* however support Sound Blaster Pro formats for 44K-mono and 22K-stereo.   It will also load non-standard sample rates if the VOC file was recorded

at one.

Brainwave frequency (Wave) function has adjustable graph for easier input of multiple frequencies over time.

CD Player has been added to play CD's for recording.   Song titles may also be entered and restored automatically on future playbacks.

DC Bias Filter added to Amplify to adjust waves skewed by a voltage.

New waves can be made at any valid sampling rate, by use of the new **Custom** option.

Monitor Source meter is in much better *real time* since it now does not depend on the selected buffer size.

Are You Sure? Dialog box added to Silence button.

New settings option for Save/Save As interpretation of the diskette icon.

Digital Delay transformation added to easily spatially locate sounds to the left or right, or for special effects.

New Setting for viewing mode of Dots (original) or Lines (dots of sample are connected by lines).

Color DIBs can be added to the .WAV file format.

Cue list and Play list supported for looping and custom play order.

New and Improved Pink noise source.   (Ideal for use with the Wave function).

When opening multiple instances of the same wave, different names are given to each instance.

Fixed an annoying bug when working with large files that caused the green slider to work inappropriately.

Fixed a major bug having to do with the progress meter.   You can now go on to do other things while Cool works in the background on long operations.

Added automatic conversion of 16-bit audio to 8-bit while playing for boards that do not support 16-bit audio.

New Instance added to File Menu

Wave files may be Dropped into the editor from the File Manager, or other applications supporting Drag and Drop.

Cool-Edit is sizeable.   You can make your window big or little or whatever.

1.27 Double-clicking on slider bar allows entry of starting and ending samples.

1.27 Pre-compute trig tables added to Settings.

1.27 Undo function added.

1.28 File -> Close added.

1.28 Customizable presets added to a variety of wave functions.

1.28a Spectral View and FFT filtering added.

1.28c Equalizer changed to Quick Filter, since FFT Filter does true, precise filtering.
Graphs have more flexibility with points.
Cue ranges can be associated with keyboard keys, so waves can be played from the keyboard.
1.29 Edit Left/Edit Right for cutting, pasting, and copying only one channel of a stereo wave.
Adjust sample rate added.

1.29a Selectable wave devices (handles multiple sound cards).

1.29b Custom file filters (WAV, VOC, PCM, and AIFF currently).   See API to build your own!

1.30 Find Frequency is now a graph of the frequency spectrum.
Fixed the remaining known bugs, and completed spec for building custom file filters.

1.30a Found more bugs: fixed Undo for Paste (special/loop/regular), and Undo Delete.
Fixed Re-Open on PCM files (would hang).
Added Undo for Music.
Added presets for Envelope and FIlter.
Added initial/final settings for overtones in Generate Tones.
Revised Music dialog by adding Constant Duration checkbox to replace Adjust Duration.

1.30b Yah, I know... There weren't supposed to be any more bugs.   But I found some:
Using FFT functions sometimes crashed (like FFT, Frequency Analysis, Spectral View, or Music).   That has been fixed!
Undo paste when Hilight After Paste was not checked didn't work.   It does now.
File filters have been modified to allow any format to be broken up into multiple files.   For example, the WAV file format can have several file filters for different compression schemes.
IMA/DVI ADPCM compression filter added to package (dvi.flt), which allows compressing of 16-bit audio downto 4, 3, and even 2-bit (with noticeable loss in quality at 2-bit).

1.31 Added the Scripts feature, which allows Cool to remember everything you do, and will let you replay it.   One script file can contain many individual scripts.
Wasn't loading from command line.
Fixed memory hog/crashing bug in Generate Tones.
Added customizable Toolbar by modifying cool.ini file.
Added play bar to see what part of the wave is currently being played
Added **Autocue** button to Play List to play a single play list item at a time.
Trim is now undoable.   Please turn off the Undo feature whenever you are doing really really large operations if you want Cool to run faster, since it will not have to save Undo information.
Added Save Selection to save the currently highlighted portion of the wave to disk.
Improved method for stretching while preserving pitch or tempo for cleaner sounding waves called **Interval Overlap**.   See Stretch.
Improved 'time left' reporting when recording.
Increased maximum file size from 248 megs to 536 megs.

Thanks go out to:

Darrell Plank for suggesting the usage of MS ADPCM wave format
Curits Palmer for directing me to the ADPCM source code
David Norris for suggesting the Transform->Silence and File->Save (not Save As) options, as well as the DC Filter option for amplifying signals that aren't perfectly "on the line".
Mike Rozak for suggesting the linked list approach, and providing some FFT code.
James Blackwell for suggesting Re-Open
Jarrod Beckman for letting James Blackwell use his machine, and for suggesting the New Instance function.

Ronald Mendoza for suggesting the Vocal Cut feature found in the Channel Mixer.
Ted Tatman for regular feedback on features and functionality.
Other Cool Fans who made suggestions and found bugs here and there.
All those who downloaded Cools 1.05 to 1.21 for your patience in waiting for this version!   I hope it was well worth the wait :-)

Other Acknowledgements
Cool Edit uses Microsoft Coropration's CTL3D.DLL dialog library, which Microsoft has agreed is a publicly available library and can be use without restrictions.   Please note that Microsoft does not support this program in any way, and are not liable for anything that may happen.

All format types have been reproduced from publicly available documentation.   I make no claims that they are in any way correct, and as such, may not be compatible with other interpretations of the same format.   It is up to you to first test to see that when saving audio files in a particular format, that the file generated is compatible with other systems that read the same format.

Common Procedures (How To...)

If you are asking "How do I...?" then you should find (some of) the answers here!   (Even though it is a terribly small list).

HOW DO I:

> **Convert cleanly from one sample rate to another**
> **Open a file for editing?**
> **Scroll wave left or right?**
> **Create sound effects?**
> **Mix multiple waves together?**
> **Copy a Mono wave to only one channel of a stereo wave?**
> **Spatially locate a mono sound to the left or right?**

Opening Waveform Files

There are a number of ways to get a waveform into Cool-Edit for editing.

- Choose **File -> Open** and find the wave file from the dialog box.

- Double-click on a waveform file from the File Manager (if you have chosen COOL as you editor of choice from the **File -> Associate** dialog).

- Click on a waveform file in the File Manager, and drag it into Cool-Edit.

- Copy a waveform from any wave editing program, then choose the **Edit -> Paste** option, or press Shift+Insert while in Cool-Edit.

- Create a new Cool-Edit program item in the Program Manager.   In the Command Line field, enter the path to the wave file to open after the program executable name.   When this icon is double-clicked, Cool-Edit will automatically load the wave file you specified in the command line, and play it.

- Choose **File -> New** to create a new waveform of the appropriate sample rate.   You may now press **Record** to record a new wave to be edited.

Converting Sample Rates

There are various methods for converting sample rates.   One way is to simply copy the waveform you wish to convert, open a new instance of Cool, and say New.   Choose the new sample rate, bit rate, and so on.   Then Paste.   When Cool Pastes samples of a different rate, it converts by doing a linear interpolation of the points inbetween when upsampling (converting to a higher sample rate), and it just picks out the samples needed when downsampling.   This method is quick and dirty, which means it is fast, but is not the cleanest method, and you will get some artifacts and "ringing" depending on the sample.

A slightly cleaner method is to use the Stretch function (stretch To cleanly convert, you must run a filter over the sample, and use the Stretch function (in Preserve Neither mode).   If you are downsampling (eg 22K to 11K), you would compress the wave to the ratio of the destination sample rate over the source sample rate (eg 11K / 22K = 0.5 = 50%).   Then you would adjust the sample rate (Edit->Adjust Sample Rate) to your new rate, keeping the bit size (8 or 16), and the number of channels the same.   The stretch function uses a weighted average approach when downsampling, which is cleaner than just pasting the sample in.

The best method, for downsampling or upsampling without noticing any difference in quality (in fact, upsampled waves sound better than the original at a lower sample rate), you must use the filter, which can be very slow.   The rest of this article explains how to use the Filter and Stretch functions to upsample and downsample.

If you are downsampling (converting from a higher sample rate to a lower one), you must Filter the wave first to eliminate any frequencies above the destination rate's Nyquist frequency.   The Nyquist frequency is a frequency exactly one half the sample rate.   For example, to convert a 44.1K sample to 22K, you must filter out all frequencies above 11K.   To do this, choose Filter.   Select 'Passive' as the filter type, and make the graph such that all frequencies above one half the destination's sample rate are set to 0%, and all frequencies below are set to 100%.   If you are cutting the rate exactly in half, as in our example, you may use the pre-defined filter "Low Pass 1/2", which does just this.

Once filtered, use the Stretch function to compress the wave.   The ratio of compression is equal to 100 * destination_sample_rate / current_sample_rate.   In our example, this would be 50.   If you were downsampling from 44.1K to 11K, this would be 25, or 32K to 22K, this would be about 68.8 (68.745% for 32075, or 68.906 for 32000).
Finally, choose Edit->Adjust Sample Rate to tell Cool what the new sample rate is.

If you are upsampling (converting from a lower rate to a higher one), you must first use the Stretch function.   The ration of expansion is equal to 100 * destination_sample_rate / current_sample_rate.   For example, if we are converting from 22K to 44.1K, this would be 200.   The stretching will do a weighted averate, which results in a linear interpolation of the samples to generate the "in between" samples.   This interpolation adds alot of high frequency noise, which can simply be filtered out in the last step.

Change sample rates by choosing Edit->Adjust Sample Rate to tell Cool what the new sample rate is.   This should be done before filtering, so the filter knows what the current sample rate is.

Choose Filter to filter out all frequencies above the Original wave's Nyquest frequency (or one half the original sample rate).   In our example, we would filter out everything above 11K, since that is one half of 22K, or original sample rate.   All frequencies below 11K should be set to 100%, while all frequencies above 11K should be set to 0%.   If you were upsampling from 8K, you would set all frequencies below 4K to 100% and above 4K to 0%.   The filtering will take the linearly interpolated samples, and smooth out all the jagged edges.   The result is a sample that now sounds better than it did at it's original sample rate.

Your sample should now be cleanly converted to the new sample rate, and sound spectacular, with no ringing, or noise artifacts.

Mono to Stereo Conversion

You can copy the wave at it's current volume directly to one channel or the other.

- Copy the wave in question to the clipboard by highlighting it and choosing **Edit -> Copy**.
- Go to the stereo wave in question, and uncheck **Edit Left** or **Edit Right** in the **Edit** menu.   Now paste the wave with **Edit -> Paste**.   The wave will be pasted to the channel that remained checked in the menu.

or

If you wish to place separate waveforms on each channel of a stereo wave and mix at different volume levels, you can use the **Edit -> Paste Special** option.

- Choose **File -> New** and create a new stereo wave of the sample rate you wish.
- Open a new instance of the program, and open the mono wave you want to place on the left channel. If you want to place a stereo wave, you may wish to use the Channel Mixer to mix both channels at 50% first.
- Highlight the section you wish to place on the left channel, and choose **Edit -> Copy**.
- Select the new blank stereo waveform, and choose **Edit -> Paste Special**.   Make sure Overlap is checked, looping is turned off, Lock L/R is turned off, and the left volume is 100% while the right volume is at 0%.
- To do the right channel, copy the section as you did for the left.
- Place the cursor at the start of the new waveform, and choose **Edit -> Paste Special**.   Change the volume levels so that the left volume is 0% and the right volume is 100%.

Spatial Location

Spatially locating a mono sound makes the sound appear as if it is coming from the left or the right when listened to with stereo headphones, even though the actual volume levels for left and right of the wave are identical.   What happens is one channel is delayed a few milliseconds.   When a sound reaches one ear a few milliseconds before reaching the other ear, the brain interprets the delay as hearing the sound coming from the direction of the non-delayed signal.

There are two functions that can do this: **Delay** and **Wave**.   Use **Delay** to place a sound anywhere from left to right, and **Wave** if you wish to have the sound source oscillate between left and right.

- First make sure the signal to be delayed is mono in stereo format.   That is, both left and right channels are identical.   With a mono source, copy the wave, create a new stereo wave, and paste. With a stereo source, use the channel mixer so that both channels get mixed at 50% for each channel.
- Highlight the entire selection you wish to spatially locate.   Choose **Transform -> Delay** or click the stopwatch icon.   Choose a delay of up to 2 ms.   Delaying one channel will make the sound appear as if it is coming from the other channel.   You can use the preset for Spatial Left or Spatial Right as well.
- If you want the source to move left to right, choose the **Transform -> Wave** function.   Enter the cycles per second (eg 0.1 = left to right and back in 10 seconds) in the initial and final boxes (the same value for both boxes).   Choose the center position (in the center, left, or right) using the Centering control.   Choose an intensity of about 50 or so, greater intensities mean the sound goes further left or right off of center.   Also check the Musical Source box if your sound is not noise related.

Creating Sound Effects

Two types of sound effects can be created using Cool-Edit: Noise based, and Tone based.   To create a sound effect, you must first generate some noise or tones to base your effect upon.   The basic method is to create a few seconds of tones or noise, then use the transformation functions to manipulate the wave and create the desired effect.   With these to sound sources, noise and tone, practically any effect can be created.

Noise Effects:
>   Waterfall, wind, and rain
>   Thunder, snare drum, cymbals, jet engines
>   Fantasy sounds such as time tunnel vortex, etc.

Tone Effects:
>   Siren, pipe organ, piano, and other musical instruments
>   Space ship sounds, whining, whistles, etc.

- First generate a few seconds of noise or tones using the **Generate** functions.
- Experiment with the different settings (eg white noise, pink noise, overtones, etc.
- Add some silence to the end of the sample by clicking the cursor at the end of the wave, and choosing **Generate -> Silence**.   This will give some room for transformations that "bleed over" such as echo.
- Try effects such as **Flanging**, **Filter**, **Quick Filter** and **Special Efx** with noise sources, or **Stretch** and **Echo** with tone sources.
- Try reversing, or copying and loop pasting portions of the wave.
- The possibilities for sound effects are endless, and is only limited to your imagination!

Function Presets

Many of the functions have presets available for easily recalling your favorite settings.   New presets can be added at any time.   All preset information is saved in the file cool.ini, usually in your Windows directory.

Double-Clicking on any preset will instantly set all controls in the dialog box to that preset.

Whenever you have settings you would like to keep, you may enter name for your settings, and press **Add**.

To remove a preset from the list, choose the preset, and press **Del**.

If all presets are removed, the default presets will automatically be reloaded.

If you are finding that your presets are not being saved the next time you use the function, check to see that the cool.ini file is not more than 64K in size.   It can grow this large if there are many CD titles in the file from the CD player.   If it is too large, remove some of the CD titles, and sorry for the inconvenience of Windows not allowing an INI file of more than 64K in size.

File Filters API

File filters (the *.FLT files you see bundled with Cool Edit) are actually separate DLLs (Dynamic Link Libraries) that contain the code necessary to read and write various file formats.   The functions that the library must contain are listed here, as well as the source used to create the PCM file filter, which can be used as a starting point for building your own.

If you do not have any .FLT files, you will not be able to load or save waveforms!

filters.h
myfilter.def
myfilter.mak
myfilter.c
API description

If you are seriously considering writing a file filter, and want more information, sample source code, or whatever then please write me.   Also, if you write a filter for a nifty new file type that is not currently supported, then please send me a copy!   I'll include it in the next iteration of the program.   And I'm sure I'll think of some other benefits when the time comes.

Undo

If the Undo function is enabled, (see <u>settings</u>), you will be able to back up one step if you make a mistake. Undo information is automatically saved in a temporary file called ~NDO*nnnn* in your temporary directory before you do anything that will change the waveform being edited.   This could slow down wave operations on very large waves.

You will be warned if there is not enough disk space to save the Undo information before continuing any wave operation.

Closing a Waveform

Close will close the wave currently being edited, and show the initial startup information in the status boxes.

When a waveform is closed, the associated temporary file is removed, thus freeing up the hard disk space that was being used previously.

Scrolling Wave

The green and/or black bar above the waveform indicates which portion of the entire wave is being viewed at that moment.   When **Zoom In** is chosen, the bar gets smaller, since the portion being viewed with respect to the entire wave is smaller.

You may click, and drag the green bar at any time to scroll the portion being viewed left or right at any time.
clicking in the slider bar region will bring up the Viewing Samples text entry box, to type in starting and ending samples.

Mixing Waves

Waves of any sample type can be mixed together into a resultant wave of any sample type.   This means you can mix a mono 8-bit 11K wave together with a Stereo 16-bit 22K wave without any problems.

- Create a new instance of the program by choosing **File -> New Instance**.
- Choose **File -> New** in the new instance to choose your final sample type.   You may generate a few seconds of silence to give yourself a workspace.
- Mix all your waves into the new instance one at a time by highlighting the portion you wish to mix, and selecting **Edit -> Copy**.   You may create more new instances and open waves into them if you wish.
- Position the cursor where you wish to place the start of the copied waveform, and choose **Edit -> Paste Special**.   Make sure the Overlap box is checked.   Choose the mixing volume, and press OK.
- Do this for each wave you wish to mix.

Edit menu

The file menu displays all the waveform editing options.

Edit Menu Options:

Undo
Copy
Cut
Paste
Paste Special
Edit Left
Edit Right
Delete Selection
Trim
Viewing Range...
Adjust Sample Rate...
Byte Swap...
Spectral View / Waveform View

Reverse
Quick Filter
Amplify
Echo
Flange
Wave
Channel Mixer
Special Efx
Stretch
Music
Envelope
Delay
Filter

Transform menu

The file menu displays all the waveform transformation options.

Transform Menu Options:

> **Reverse**
> **Quick Filter**
> **Amplify**
> **Echo**
> **Flange**
> **Wave**
> **Channel Mixer / Inverse**
> **Special Efx**
> **Stretch**
> **Silence**
> **Music**
> **Envelope**
> **Delay**
> **Filter**

Generate menu

The file menu displays all the waveform generation options.

Generate Menu Options:
Generate Noise
Generate Silence
Generate Tones

Options menu

The file menu displays various additional options.

Options Menu Options:

> **Loop / Play Toggle**
> **Monitor source**
> **Settings...**
> **Info...**
> **Frequency Analysis...**
> **CD Player**
> **Cue List**
> **Play List**
> **Cool Scripts...**
> **Select Wave Device**

The **View Indicator** is the green bar above the waveform being viewed which changes size depending on the level of Zoom.

The **Toolbar** is the row of icons just below the menu for quickly accessing commonly used functions.

Quick Help messages appear below the toolbar to give a brief description of the function associated with the tool icon.

Navigating Cool Edit

Commonly used functions are represented as icons in the <u>Toolbar</u>.
-   Clicking, and holding on any of these icons will bring up a <u>Quick Help</u> message describing the function in more detail.

Waveforms are displayed in green.
-   Any portion may be selected by clicking on the waveform, and dragging left or right.
-   Holding the **SHIFT** key while doing so will increase the currently selected portion.
-   **Double-Clicking** on the waveform will select the entire visible wave.

The <u>View Indicator</u> depicts which portion of the entire waveform is being viewed in the workspace below.
-   **Sliding** the green bar (when present) left or right scrolls various parts of the waveform below into view accordingly.
-   **Double-Clicking** on the <u>View Indicator</u> will bring up a <u>Quick View</u> screen to allow direct entry of starting and ending samples for the viewing range.

The **Yellow Arrows** indicate the point of insertion.   Clicking anywhere in the workspace will move the point of insertion to the mouse.

Choose **Zoom In** to expand the selected portion to full width.   **Zoom Out** will give a larger view of the waveform, while **Full View** will display the entire waveform in the workspace.

The **Play** button will play the portion of the wave that is currently being viewed, or the portion that is highlighted.
The **Record** button will start recording at the current insertion point.   Any waveform data after that will be recorded over.
Use the **Stop** button to end waveform playback or recording.

The **Record Level Meter** below the Play/Record buttons will display red when a waveform is being recorded, and a real-time display of the input when Monitor Source is chosen.
-       **Double-Clicking** on the Record Level Meter will start and stop monitoring.

The **Time Display** windows show the current Starting and Ending points of the current selection, or portion being viewed.
-   **Double-Clicking** on these windows will toggle the display between time in seconds, and in samples.

 The **Wave Format** window displays the format of the wave in Channels, Sample Rate, and Bits Per Sample.
-   **Double-Clicking** on this window will bring up the Adjust Sample Rate dialog to change the waveform's characteristics.

When a wave is being played via the **Play** button, or automatically via a keypress (see <u>Cue List</u>) a vertical bar shows the current playing position.

If a valid audio CD was in the CD-ROM drive, and the [MCI] CD Audio extensions are loaded, <u>CD Player Controls</u> will appear at the bottom of the window.

Quick Keyboard Shortcuts
Delete   Delete
CTRL+C   Copy
CTRL+Insert   Copy
CTRL+X   Cut
SHIFT+Delete Cut

```
CTRL+V        Paste
SHIFT+Insert  Paste
CTRL+T        Trim
```

Opening a Waveform

Various waveform data types are supported.   Cool-Edit will convert the waveform type to its own internal linked list type for faster editing, and larger edit file size.   The wave can subsequently be saved in any of the supported formats you wish.

Each wave type supported is embodied in a filter (*.FLT) file.   I have provided the file filters for the following formats:

WAV - Windows PCM waveform
Standard Windows PCM waveforms consist of a RIFF WAVE header, followed by pulse code modulation waveform data.

WAV - Microsoft ADPCM waveform
Microsoft ADPCM compressed waveform format.   Waves are saved in 4-bit per channel compressed format.   Each 4-bit sample is expanded to 16-bits (or cut down to 8-bits if needed) when loaded.

WAV - IMA/DVI ADPCM waveform
This standard compresses 16-bit waves to 4-bit in a different (faster) method than Microsoft ADPCM, and has different distortion characteristics, which may be better, or worse, depending on the original sample being compressed.   This format allows for 3-bit compression as well at a slightly lower quality.   I have implemented a 2-bit version of the compression by using the index table -1,2,-1,2.   I could not find the specification for 2-bit compression, so it may not be compatible with other IMA/DVI 2-bit compressed files. I found that the preceeding index table provided the best quality.

PCM - Raw PCM Data
Simply the PCM dump of all the data for the wave.   No header information is contained in the file.   For this reason, you are asked to select the waveform sample rate, resolution, and number of channels.   By reading audio data in as PCM, practically any audio file format may be read in!   You must have some idea first of how many channels, and the sample rate.   The waveform will sound bad in different ways depending on which parameters you have mixed up.   Once the wave is loaded, and sounds fine, you may hear clicks at the start, end, or sometimes throughout.   The clicks are various header information being interpreted as a wave.   Just cut these out, and "Whoo-La"!   You have read in a wave in an unknown format!

VOC - Sound Blaster voice file format
Sound Blaster and Sound Blaster Pro voice files can be loaded.   Only 8-bit formats are supported. Maximum sample rate for mono is 44100 samples/second, and 22050 samples/second for stereo.

AIF - Apple AIFF format
Apple's standard wave file format.   Assuming you got the wave over to the PC, you can name the file with the .AIF extension, and load it using this file filter.   I have tested it on several Apple sound files, and have been able to read them in OK.   I do not guarantee the reverse will be true, but Macintoshes should be able to read in AIF files saved in this format.

Re-Opening a Waveform

This will reload the previously loaded waveform, replacing any changes that were made to the waveform being edited.   Be aware you will lose any changes you have made since the last time the wave was saved.

Editing Individual Channels

For stereo waveforms, it is sometimes handy to be able to copy from, and paste to individual channels. Normally, both **Edit Left**, and **Edit Right** are checked, meaning both channels are being edited simultaneously.   To edit only one channel, un-check the channel you want to preserve, so only the other channel is being edited.   When you do this, only the most basic editing commands are active.   When pasting, the audio data is overlapped with what is already there, since inserting only on one channel will put the stereo-ness completely out of phase.   If this is your desired effect, use the <u>Delay</u> function.

Viewing Range

Selecting **Edit -> Viewing Range** or double-clicking on the green/black samples portion bar will bring up the viewing range window.

Enter the leftmost and rightmost samples that you wish to have displayed.   Use this function to highlight a specific number of samples by double-clicking in the wave editing area after selecting the viewing range to "select all".   Remember, Select All only selects the viewing range, not the entire waveform.

New Instance

... Or "Give me another window"

Selecting New Instance will open another Cool-Edit window as if you opened another instance from the Program Manager.   This may be handy when you need a secondary window to do some editing in, and don't want to disturb the original wave.

Frequency Analysis

This will bring up an analysis dialog, containing a graph of the frequencies present at the insertion point (yellow arrow cursor), or at the center of a selection.   A Fast-Fourier-Transform is performed to determine the frequencies present.   Also, the most prominent frequency is interpolated, and displayed below.   By moving the mouse over the graph area, the frequency, and percentage (just the height of the mouse position) are displayed.

The information in this dialog is like one "slice" or line in the Spectral View of the waveform.

Check the **Normalize** box to re-scale the graph so that the highest intensity sample reaches the top of the graph.

Spectral Viewing

Choosing **Spectral View** will display waveforms by the by their frequency components.   This is a handy function for analyzing your audio data, to see which frequencies are most prevalent throughout your data.

The more abundant a frequency is, the brighter the color displayed will be.   Colors range from dark blue (next to no frequencies in this range exist) to bright yellow (frequencies in this range are very strong).

Lower frequencies are displayed near the bottom of the display, while higher frequencies are displayed near the middle on to the top.   The display is linear.   White lines on the left and right divide the display into 1/2, 1/4, 1/8, and so on.   To top of the display represents frequencies at just below the Nyquist frequency, or 1/2 the sample rate.   So if a bright spot appears near the top of the display for a signal sampled at 22K, the frequency being represented is near 11K.

Choose **Waveform View** to return to original waveform view mode.

Select Wave Device

If you have multiple sound cards, or multiple output devices (such as a sound card *and* the PC speaker), choose the input and output devices you wish to use.

If your system is equipped with MIDI devices, you may also choose the MIDI in, and MIDI out sources.

The settings are remembered in the [cool edit] section of your WIN.INI, which means if you install a new sound driver or card, Cool will not access it until you choose it from this dialog.

FFT Filtering

You may filter out undesired frequencies, or just keep certain desired frequencies by using the **Passive** mode.

Use the **Logarithmic** mode to boost or dampen frequency components.

When the **Lock** is not set, you can choose both an **Initial** and a **Final** filter.   Filtering will gradually go from the initial state to the final depending on the <u>Transition</u> settings.

The **Precision Factor** determines how accurate you want the filtering over time.   A low factor, means the filter settings will change roughly, or in chunks.   With higher settings, the filter's transitions are much smoother.   In any case, the higher the precision factor, the longer it will take to filter your selection, but the nicer it may sound.   The FFT (Fast Fourier Transform) function takes a large group of samples, and filters them all at once.   The precision factor determines how many samples from the entire group are actually saved in the final product.   A factor of two means that half the samples are saved back.   A factor of 10 means that 1/10 of the samples are saved back.   Since there can be only one filter setting for the entire group of samples, you will want smaller groups of samples if the settings are varying widely over short periods of time.

If **Morph** is checked, the transition from the initial filter settings to the final filter settings will actually "morph" from one to the other.   If this is not checked, the settings simply change linearly over time, which means if you have a spike at 10K for the initial filter, and a spike at 1K for the final filter, the spike at 10K will gradually decrease, and the spike at 1K will gradually increase over time.   If morphing is on, then the spike will "ooze" from 10K down to 1K, passing many of the frequencies in between.

Really nice effects can be heard by simply choosing the **Passive** mode, and having an initial setting with first half of the filter at 100%, and the second half at zero for the initial filter, and the right 1/10th or so at 100% with the rest at zero for the final filter.   This selects high frequencies for the initial configuration, and low frequencies for the final configuration.   To get a nice blending from high to low, choose **morph** to blend the two together by including all the frequency combinations between the two filters.   To see exactly what is happening as the filtering changes from the initial configuration to the final, choose <u>Transition</u> to view the actual settings that will be used over the duration of your selection.

This function supports <u>Presets</u>.

For best results, filter using 16-bit samples.   If your source is 8-bit, copy, create a new blank 16-bit waveform, and paste.   You can convert back to 8-bit when done.
For more accuracy in using any transformation, always use 16-bit samples.   If your board only supports 8-bit samples, you can have them converted on the fly at playback time by choosing the "convert 16 to 8 bit" mode in Settings.

FFT Transition Settings

Choose how you want your sample filtered over the time of the sample.   The left of the input graph represents the start of your sample, while the right hand side represents the end.   You can choose how your selection will be filtered over the length of the selection.   The low points represent filter settings close to your initial filter, while the higher points represent the settings close to your final filter.   All points in between are a combination of your initial and final filter arrangements.

The actual graph of the filter being used at any point can be seen below.   The filter setting shown corresponds directly to the position of the mouse in the input graph.   By watching how the filter settings change as you move your mouse up and down in the graph area, you can decide whether you want a morphing transition, or a linear transition.   Each type of transition will give different filter settings for the points between your initial and final filter settings.

Morphing is generally any technique used to transform one object into another.   In the case of filter settings, it is a way to smoothly transform one setting (represented by a graph) to another by estimating all the possible combinations of the two settings.   Over time, the first setting becomes the second.   At some point, the setting will stop looking like the initial configuration, and start looking like the final configuration.   In the filtering world, this means that frequencies between the ones selected to be filtered will also be filtered.

In going from the initial to the final filter configuration, the points in the "in between" settings are just the average between the two settings.   For example, a filter setting exactly between the initial and final would be the exact average of the initial and final filter settings.

Musical Incantations

Now you can put your clippings to music, or just harmonize a wave using a particular chord.   To choose a clipping for your sample, select the range you wish to use as a quarter note.   If no range is selected, the clipboard data will be used.   Note that the clipboard data will be filled with your sample automatically once music is generated, thus selecting music a second time will automatically use your last sample.

How to build a song
Simply drag the notes and rests you desire to the music bar above.   To sharpen or flatten a note, drag the sharp (#) or flat(b) symbol on top of the note you wish to transpose.   You can move notes up or down after they have been placed, or pick them up to insert in a new position.   To remove a note, pick it up and drop it off away from the bar.

Use the scroll bar to work on individual portions of the song at a time.   You can scroll to write a piece as long as 256 notes.

Tempo
The tempo is given in quarter notes (beats) per minute.   Your sample's length is the length of a quarter note.   If your note is longer than the period determined by the tempo, then the notes will overlap.

Key
You may choose to have your music interpreted in any of the standard key signatures.   The key of C is the standard (white keys).

Constant Duration
If chosen, all notes will be the same length as the original sample, regardless of pitch.   The operation that does this takes longer to calculate, but high pitched notes will be the same length as lower pitched notes. The Interval Overlap method is used with an overlap of 80% and an interval of 30 Hz.   If not checked, the note is created by directly stretching or compressing the original sample, resulting in higher pitches being shorter than lower pitches.

Exact Tune
Choose Exact Tune to tune your sample so that when played at A (above middle C), the frequency of your sample is at 440Hz.   If this is not checked, your sample's original frequency will be played at A (above middle C).

Chords
The triplets of numbers to the right is the chord selection box.   You can choose to make a chord out of 2, 3, or 4 notes, then choose the chord from the list.   Finally, pick up a chord object (the 3 notes on top of each other) and drop it on a note above.   The note you drop it on will be the starting note of the chord, and the other notes will automatically appear above it in the right ratios.

Clearing Chords, Sharps, or Flats
If you want to clear a sharp, flat, or chord from a note, use the faded looking quarter-note object, and drop it on the note you wish to bring back to normal.

Saving Your Songs
If you make a cool song you want to keep, give it a name in the **Song Title** box.   In the future, you can choose your song from the list of song titles that you created.   The actual song data is saved in the file SONGS.INI in your Windows directory.

Listen
If you have MIDI play capabilities, you can listen to a preview of your song before actually creating it. Play begins at the leftmost note visible on the staff, which means play begins at the position you are scrolled to, and continues on to the end of the song.   The music is played through channels 1 and 13 for

Extended and Base level compatibility.   The instrument can be chosen by typing its instrument number to the left.   You can record music played by the listen preview button.   Simply hit the record button first, then go into the music dialog and press Listen.   When the song is done, hit Cancel, and then Stop to stop the recorder.

Pink
What the heck is this button for?   Well, it automatically plays the chosen instrument through the MIDI, using pink noise as the source for randomness.   Maybe once out of 1 million tries, it may actually write a cool song?   Tempo, Octave, and Key all affect the play of "pink" music.   The only purpose for this button I have found so far, is to open the Play List first, then open the music dialog.   While listening to a relaxing soundscape from the play list, you can listen to relaxing random music at the same time. (?) (!)

Saving a Waveform

Currently, four formats of waves are supported.

WAV
This is the standard Windows .WAV file format for uncompressed waves.   There is an option to encode in ADPCM format, which encodes each channel per sample in 4 bits, giving a 4:1 compression ratio for 16-bit sources, and 2:1 compression for 8-bit sources.   Because of this, it is always best to encode 16-bit data if possible, since it does not take any more or less space after compression.   Saving waveforms in this format may take up to 10 times longer to save, or more!   This is because the optimum compressed data is being sought out for minimum distortion after decompression.

PCM
This is the raw data only, with no header.   This format is handy when converting to other formats not supported.   It can also be used when mastering compact discs.

VOC
Sound Blaster voice file format.   Waves are broken up into "blocks".   Cool tries to make sure the left channel and right channel are at the same volume at the ends and beginnings of blocks in stereo files to eliminate clicking heard with various voice file players.

AIF
Apple's AIFF sound file format.   An AIFF header is placed before the PCM sound data.

Saving the Highlighted Selection

Saving a wave selection is identical to <u>Save</u> except that the highlighted selection is saved, and not the entire waveform.

Immediately Saving a Waveform

Choosing **Save** from the **File** menu will save the current waveform back to disk, overwriting the original without confirmation.

The disk icon in the toolbar can be set to save immediately, or to bring up the **Save As** box depending on the setting in the Settings   section.

Settings

Play/Record Buffer Size
The number of seconds to reserve memory for recording and playback.   Increasing this will allow more multitasking while audio is being played, but it takes more memory.   If this value is too small, there may be too much choppiness in your recordings and playbacks.   If your recordings are getting all "chopped up", or you cannot Stop after you've started recording, increase the buffer size, or switch to a faster hard drive (Use a non-compressed hard drive for example).

Disk Icon Interpretation
The Disk icon can be interpreted as:
- Save Now (saves what you are working on under the same filename seen in the title bar without asking if you wish to overwrite the original) or
- Save As (Always brings up the Save As dialog box to enter the title to save the file as)

Viewing Mode
When zoomed in closely to a wave, individual samples can appear as dots, or as a continuous line.

Highlight After Paste
After doing any Paste operation (this includes Paste Special), you can have the inserted selection automatically highlighted, or just have the cursor at the end of the pasted selection.   Not highlighting after pasting makes it easier to do multiple pastes one after the other.

Pre-compute Trig Tables
If you are not using a co-processor, or a 486 or better machine, some functions will run faster if this is checked.   Tables for trigonometric functions are computed once, and used during the Wave, Flange, Special Efx, and Generate Tone functions.

Play 16-bit files as 8-bit
If your sound board is only capable of 8-bit audio, you can still create and edit 16-bit audio files.   When you choose Play, the audio data will be converted to 8-bit before being sent to the sound board.

Enable Undo
If this is checked, the undo function is enabled and running.   After making a change, you can choose **Edit -> Undo** to back up one step.   You may want to disable the undo feature if you know you are not going to undo something.   For example, if you are running a function on a 5 minute file, you may not want to wait while the undo information is saved.

NT Compatibility
If this is checked, then the program "locks" while the green waveform is being drawn.   With Windows NT, other apps are not paused when this is going on, and the extra code needed to "multitask" with other apps does not need to run.   If you are running Windows NT, and this is not checked, you may experience various screen redraw problems.

Cue List

A cue list is a list of time offsets into the wave file.   A cue can be either a point, specifying a cursor position, or a range, specifying a selection.   You can easily jump to a cue position in a wave by double-clicking on the position in the list, or selecting the cue position, and pressing Goto.   Cue ranges can later be arranged in a play list to be played back in any order, with a specific number of loops if desired.   A maximum of 96 cues may be entered.

<- Add
Add the currently highlighted selection, or cursor position to the cue list.   Items will be displayed in temporal order, with the earliest cue position at the top of the list.

Remove
Remove the selected cue position from the list.

Label
Short text label describing the selection.

Description
A textual description of the wave data if necessary.   Also can be used as a comment.

Goto
Goto the selected cue position, or highlight the selected range.   Double clicking a cue item acts as if Goto were pressed for that item.

The cue list is saved in the .WAV file format in the 'cue ' chunk.   Additional information about the cue position, like label, description, and length of sample, are placed in the 'adtl' list in the 'labl', 'note', and 'ltxt' chunks.

Assigning Cue Ranges To Keys
If you wish to assign any cue range you have added to a key on the keyboard, give the cue range a label of the form **KEY N**, where **N** is any key on the keyboard (capital letters only).   When you go back to editing the waveform, pressing the key will play the cue range you selected.   You can assign any portion of the waveform to any key on the keyboard.

Play List

The play list is a listing of cue ranges that can be played in any order, and looped a specified number of times.   The play list is used in conjunction with the cue list.   Maximum size of a play list is 64 entries.

<- Add Before
Add the currently highlighted selection from the cue list to the play list.   The selection is inserted before the currently highlighted play list item, or at the end if nothing is selected.

Remove
Remove the selected play list item from the list.

Loops
The number of loops to loop the selected cue range in the play list.

Play
Play the cue ranges in the order listed, looping selections if necessary.   Play begins at the currently highlighted item in the play list, or the entire list is played if [end] is selected, or there is no selection.

Autocue
Play the currently highlighted item in the play list (or the first item if nothing is highlighted), looping if necessary, and stop on the next item in the play list.   Thus, every time **Autocue** is pressed, the next item in the play list is played.

The play list is saved in the .WAV file format in the 'plst' chunk

Reverse

This does just as you'd expect. Select a portion of the wave using the mouse, and select Reverse from the Transform menu to reverse your wave.

Trim

Trim is the exact opposite of delete, which means everything is deleted *except* the portion that is selected. Only the selected portion is kept.   This is handy to quickly pick out the part of a recording you want to keep.

Use CTRL+T to quickly trim the selected portion of the wave.

Editing Channels of Stereo Waveforms

When fully implemented, you will be able to make edits (copy, paste, etc. and so on) to only one channel, completely independent of the other.

Digital Delay

Delay either channel up to 50 milliseconds with the option to mix in the original signal with the delayed signal.   Great for effects such as spatially locating a previously mono wavesource to the left or to the right, so that the sound will appear to emanate from that direction when listened to with stereo headphones.   Delays of longer than 50 ms may be entered for creating a single echo.

Delay (ms)
The actual amount of time to delay the channel in question.

Mixing
You may choose to have the resultant wave the delayed signal, keep it the original signal, or mix the two. A value of 50 will mix the two evenly.

Invert
The delayed signal may be an inverse of the original if this is checked.   More special effects!

This function supports Presets.

Tunnel Preset
The Tunnel preset can be used with mono as well as stereo waveforms.   The settings provided give a nice tunnel/tubular effect.

Spatial Left Preset
If a mono wavesource was converted to stereo (so that the left and right channels are the same), then choosing this will make the sound appear as if it is coming from the left, since the right channel is delayed just enough so your brain interprets the sound as coming from the left.   You must use headphones to hear the effect.

Spatial Right Preset
The same as Spatial Left, but it locates the sound to appear as if it is coming from the right.

Stretch

You can choose the initial stretching percent (less than 100% will compress the wave), and the final stretching percent.   By having two separate values for start and end, the wave can be stretched linearly from one ratio to another.   This gives the effect of slowing down and speeding up, or raising and lowering pitch.

Preserve Pitch
Lower percentages will slow down the tempo, while higher ones will increase the tempo.   The pitch remains the same throughout.

Preserve Tempo
The tempo or speed of play will remain the same.   The only thing affected will be the pitch.   Higher percentages will lower the pitch, and lower percentages will increase the pitch.   Try using differing initial and final percentages to raise and lower the pitch without affecting the tempo.   First the selection is adjusted, preserving the pitch, then the selection is squeezed or expanded, with no preservation.

Preserve None
The tempo will slow, while at the same time the pitch will lower if percentages above 100 are used.   For lower percentages, the tempo will speed up and the pitch will increase.

Transpose
When the number of desired keys up or down are chosen, the numerical values for transposing musically are entered into the stretch sliders.   For example, to make your sound as if it were the next key higher (if played on a keyboard, and black keys included) choose 1# for 1 sharp.   The 'b' values will flatten or lower your sound.

Interval Overlap
When preserving pitch or tempo, the waveform must be elongated or truncated smoothly and preserve as much of the original information without adding noticable distortion.   This is one of two methods that can be used to achieve this.   The amount of distortion introduced is not dependent on the type of sample (eg music or speech).   When stretching or compressing, the appropriate chunk from the original wave is output to the transformed wave, and overlapped with the previously transformed chunk.

The **interval rate** determines the size of a 'chunk' of audio data.   Interval rates will become an audible *hollow* sound when large rates (above 50Hz) are used.   If the rate is too low, echoing will be very noticable when raising pitch, or slowing down tempo, or chopped sylables will be noticable when lowering pitch, or speeding up tempo.   Values of 20Hz to 40Hz usually produce good results.

The **overlapping** determines how much of the previous chunk is overlapped with the current chunk.   The overlapping can produce a *chorus* effect.   To reduce the chorus effect, lower the overlapping percentage.   When the overlapping is reduced, a *choppiness* to the sound may appear.   Adjust the overlapping to your taste to strike a balance between chopiness and chorusing.

Zero Splicing
This is the second method of preserving pitch or tempo.   Less distortion is noticed when using samples that were derived from one source, and contain a relatively low number of fundamental frequencies.   For example, a single instrument, or speech will work OK, while an orchestra or music will not work as well.   When stretching or compressing,   the wave is broken up into chunks that begin and end when the waveform crosses zero, or the midpoint.   Chunks are repeated, or thrown out depending on the compression ratio and the cutoff frequency.   Chunks smaller than the **cutoff frequency** will not be thrown out or repeated.   Try cutoff frequencies between 50Hz and 300Hz for best results using this method.

This function supports Presets.

Raise Pitch Preset
This will adjust the settings so that the pitch of the sample will be raised by 1/4 octave.

Lower Pitch Preset
This will adjust the settings so that the pitch will be lowered by 1/4 octave.

Speed Up Preset
This will speed up the sample, so talking will be faster, but the pitch will be unaffected.

Slow Down Preset
Slows down the sample.

Note: An alternative method of stretching while preserving pitch or tempo is to do an FFT, and modify the amplitudes and phases of each frequency in the frequency domain, and then do the inverse FFT to translate back.   This method would be very complex, and be very very much slower than the methods currently being used.   If you want to experiment some, try the /i Shift option in the Filter function does this in a very crude way, by just shifting the frequencies up without regard to their phase.

Compact Disc Player

If you have the **[MCI] CD Audio** driver loaded, you can control the CD player with a standard set of control icons.   You can also name your CD, and the individual songs as well for display the next time you use the same CD.

Tracks List
Click on any track number to start playing that track.

Time Readout
Displays the current time in minutes:seconds into the current track.

Title Display
Displays the title of the CD.   The title defaults to the length of the CD.   You can type in the proper title of the CD here.   If a track is currently being played (if the track number is highlighted in the track list), then the title reflects the title of the current song.   As you are entering song titles, you may use the TAB key to jump to the next song to easily enter the titles for the entire CD.   Titles are saved in the file COOL.INI.

Currently, only up to 64K of titles data can be stored (the limit for any INI file), so be aware that you may not be able to enter in data for every CD you own.   It may be better, if you have tons of CDs, to just enter in the title of the CD, and only enter in the song titles of those you listen to the most.   If there is a great need to allow more than 64K of titles data, I will work around the limitation to allow it...

[Stop]
Square Box: Stops CD playing.   Play will resume at the start of the CD.

[Pause]
Vertical Bars: Pauses the CD.   Play will resume at the same location.   This button will turn into a **Play** button when pressed so that play can be resumed by pressing it.

[Play]
Right Arrow: Starts the CD either at the beginning of the disk, or at the paused location.   This button will turn into a **Pause** button when pressed, so that play can be paused by pressing it.

[Scan Back]
Double Left Arrows: Rewinds the CD 10 seconds.

[Scan Forward]
Double Right Arrows: Forwards the CD 10 seconds.

[Mark]
Red X: Mark the location currently being played.

[Goto Mark]
Arrow to Red X: Go to the location that was marked earlier.

[Eject]
CD Ejecting: Spit out the CD if that is possible on your player.

Selecting **Exit**, or double-clicking the system menu box will close the program.   At close time, any temporary files that were created will be removed.

RIFF Info

Extra information can be included in your .WAV files using the RIFF LIST INFO and DISP type 1 formats. This information should (depending on future wave editors) stay with your sound through it's lifetime. Wave editors should preserve some of the fields you see here. Be sure to put proper information in it's place!

Display Title
This should describe the sound, or text (if there are words in the wave). This field should be as short as possible, since it will be displayed in OLE objects and the like.

Icon
Any DIB or BMP file can be inserted, but preferable a 32 X 32 16-color would be best. The Media browser uses this size to display a picture representing the sound. Other OLE compatible applications can use the above display title, and/or the bitmap to represent your waveform.

Original Artist
The one who created the sound initially. Examples are: Beatles, Pat Sejek, Fred Flinstone

Name
The title of the wave. This is your chance to put a name with your audio "artwork". Examples are: Thunderstorm At Night, Forest Stream

Genre
The Genre of the original work. With audio, let's try things like musical classifications, etc. Examples are: Cartoon Voice, New Age, Instrument

Key Words
In the future, sounds may be searched for by key words. Please separate key words by a semicolon followed by a space. For Example: Violin; Hayden; Johann Strauss

Digitization Source
Where was the sound digitized from. A tape deck, CD, or maybe directly from a microphone? Maybe describe the board used here too, like Sound Blaster Pro, or MediaVision. For Example: DDD CD to MediaVision Pro 16

Original Medium
Where did the sound come from originally. Examples: Live Band, Flute, Moog, Voice

Engineers
Store the name(s) of the engineer(s) who worked on the file, or edited the file. Please separate names by a semicolon and a space. When a new person edits the file, they can add their name to the list. For example: John Cravitz; Fred Millstone

Digitizer
Who is the technition that did the actual digitizing? They should put their name right here.

Comments
This is for making any comments you wish. Feel free to include any special effects or enhancements you made to any preexisting waves so that the editing history can be tracked. Please do not use any line returns. End each sentence with a period. For Example: It took me 12 hours to get this recording right. John added echoing effects using Cool Edit.

Subject
This Describes the content of the file. Feel free to include a description of the instruments used, where

someone can find the song recorded, etc.   Line returns are OK, and are created by pressing Ctrl+J.
Sometimes copyright information is placed here as well.   For Example: The shakuhachi of
Japan.<Ctrl+J><Ctrl+J>The shakuhachi was developed in the 15th century from a Chinese end-blown
flute, called the chiba.

Source Supplier
The name of the person, or organization who supplied the original source material.   Let's use this field for
the names of record companies, or whoever supplied you with the source.   Examples: MCA Records,
Ann Wilson (if recorded live)

Copyright
Any copyright information for this file should go here.   Example: (c)1992 G. Willikers Corporation. All
rights reserved.

Software Package
The software used to digitize and edit this file.

Creation Date
The date that the subject matter was created.   The date should be in the format yyyy-mm-dd, using '0' as
a place holder in single digit values.   For example, if the date the original recording was made was July
30, 1988 then it would be written as: 1988-06-30

Embedding Bitmaps in WAVs

Any DIB or BMP bitmap can be inserted into the wave file.   The bitmap will only be saved if the file is saved as WAV.   Icon sized bitmaps (32 X 32) are preferable, since they are easier to display by other OLE aware applications.   OLE applications may represent your wave file using the bitmap you select.

Cut

Cut will copy the selected range to the clipboard, and remove it from the waveform being edited.   Once in the clipboard, it can be pasted or special pasted to waveforms, or pasted to other applications that support the CF_WAVE cliIpboard format.

Use CTRL+X or SHIFT+Delete to quickly cut the selected wave.

Envelope

By using an envelope, you have great control over which parts of your wave are amplified, and by how much.   A point at the top of the graph is 100% (normal) amplification, and at the bottom is no amplification (silence).   This function is handy when modifying Tones generated with this program to create more realistic sounding instruments and effects.

Amplification
Adjust this value to amplify more than 100%.   This value changes the values represented by the graph.

Click in the graph area to add control points.   You can drag control points up and down.   Drag a control point off the graph area to remove it.

This function supports Presets.

Generate Silence

Enter the number of seconds of silence desired to generate that amount of silence at the insertion point.

Pretty simple, eh?

Silence

Choosing this will silence out the selected material.

Just about as simple as <u>Generate Silence</u>, huh?

Generate Tones

Generating tones is a great way to provide a base sound to create spectacular special sound effects.

Base Frequency
The main frequency (F) that will be used for sound generation.

Modulate By
Enter the variation in frequency you wish to hear.   For example, choosing 100 will oscillate the tone being generated between 50 minus and 50 plus the base frequency.

Modulation Frequency
This is the rate (times per second) at which the frequency modulates.   Entering a value of 10, for example, will generate tones that warble at the rate of 10 times per second.

Flavor
Choose the type of waveform to use.   Sine waves sound soft, while Triangle and Sawtooth waves are sharper.   Each flavor has a particular sound unique unto itself.

Stereo Phasing
The left channel can be out of phase with the right channel.   If you wish this, choose the amount of phase shift here.   A value of 180 will be completely out of phase.

Duration
This is how many seconds of tones you wish to produce.

Frequency Components
You can choose up to 4 overtones, and mix them at any proportion.   Adding overtones gives truer musical effects, such as piano and pipe organ.   The 2F slider, for example, will mix in a frequency that is exactly twice the base frequency.

Lock
If checked, the overtones generated are constant -- they do not vary over time.   Uncheck this box to dynamically change the proportion of any overtone over time by choosing the inital and final proportions.

Initial and **Final**
You can choose to adjust the initial or final overtone proportion settings when the overtones vary over time.

Flip
Check (or uncheck) Flip to swap the initial and final settings.   This is very useful if you wish to generate another tone *after* the last tone that was just generated.   Whatever overtone settings the last tone had as the final settings can be used as the initial settings on this tone by clicking Flip.   All you need to do is click Final, and enter new final overtone values.   This way the tones will flow together smoothly.

Volume
The Volume sliders are used to select how intense the tones will be in each channel.   Both channels can be controlled independently when generating stereo tones.

This function supports Presets.

Experiment with all the settings for various wild effects.

... and Have Fun!

Paste

paste will insert the wave from the clipboard at the current <u>insertion point</u>, replacing any waveform data in the <u>selected range</u>.   If the format of the waveform data in the clipboard differs from the format it is being pasted into, it will be converted accordingly before pasting occurs.

Use CTRL+V or SHIFT+Insert to quickly paste the waveform that is on the clipboard.

Delete Selection

Once a range is <u>Selected</u>, it can be removed by choosing this option.   The deleted portion is *not* copied to the clipboard.   It is gone forever.

Quick Filter

The 8-band quick filter allows one to customize to suit most filtering needs.   The "equalizer" works pretty much the same as a standard audio equalizer does.   Except that the bands are not the same as you would expect.   The highest frequency band *will* increase or decrease the high end, but it will also increase frequencies all the way down to the lowest as well, but it will increase the high frequencies more than the low ones.   The effect is close to an equalizer, but not quite.   Basically, this is a handy function for changing the tone of your sample (such as noise) to make it more pleasing to the ears.

Equalizer Bars
Adjusting these increases or decreases the frequency component specified beneath the bar.

Volume Bars
The final volume after equalizing may be adjusted to suit your needs.   Checking the **Volume** checkbox will lock the left and right scroll bars.

Flat
This simply places all equalization values plus volume adjust at 100%

Lock Initial/Final
When locked, the entire selected range is equalized with the setting shown.   If unchecked, the initial and final equalization settings may be adjusted, so the selection can smoothly glide from the initial equalization setting to the final setting over the range selected.

View Initial
When Initial/Final is not locked, choose this to select the initial equalization settings.

View Final
When Initial/Final is not locked, choose this to select the final equalization settings.

This function supports Presets.

To produce a semi low-pass filter, set the higher frequency scroll bars to zero to cut out higher frequencies.   A high pass filter can be done in the reverse fashion, by zeroing out the lower frequencies.

Very interesting effects can be made by selecting widely varying initial and final equalization settings.

Note: Setting the lower bands to very high values can, and most times will, result in clipping if the volume adjustment is not turned down.

For serious filtering, see Filter.

Copy

Waves can be copied to the clipboard to be pasted later using this program, or any other application that supports the CF_WAVE clipboard wave format.   Before copying, the portion that is to be copied must first be selected.

Extra large waves may not be able to be copied.   If this happens, try breaking up the selected range into smaller portions and copy/paste them in multiple steps.

Use CTRL+C or CTRL+Insert to quickly copy the wave.

Amplify

Amplify will increase or decrease the volume of the selected sample.

Initial Amplification
This is the amplification that will affect the beginning of the selection.   Choose a separate final amplification for fading up/down effects.   An amplification value of 100 will keep the signal unchanged.

Final Amplification
This is the amplification that will affect the ending of the selection.   Setting both the initial and final amplifications to the same value will amplify the entire selection the same amount.

Lock Left/Right
Left and Right channels may be amplified at separate values.   If the Lock is checked, then the scroll bars for the left and right channels are locked to the same value.   Effects such as panning from left to right can be achieved by choosing separate values for the left and right channels.

DC Bias Adjust
Adjust the waveform so it is centered on the center line (0 %).   If samples are recorded with a DC Bias, they will appear to be above or below the center line.   They must be centered before doing other waveform transformations, and choosing this will center the wave properly.   To skew the entire selected waveform above the center line, enter the percentage to move the waveform up in the adjustment box. For example, 50% will move the entire waveform up half way, and a -50% will move it down half way.

Normalize
Pressing the Normalize button will calculate the greatest amplification for the sample that will *not* result in clipping when set to 100%.   If the left and right scroll bars are not locked, separate left and right values will be computed, potentially amplifying one channel more than the other.   To normalize to less than the maximum range, enter the percentage of maximum to normalize to.   For example, choosing 50% will compute values needed to amplify the file no more than 50% of maximum, resulting in a 3dB diminuation from maximum output.   If two sounds normalized to 50% are overlapped, the resultant wave is guaranteed not to exceed the boundaries, and will not clip.

To achieve a fading in effect, choose an initial amplification of 0, and a final amplification of 100.   For fade outs, do the opposite by setting the initial to 100 and the final to 0.

This function supports Presets.

Fade In Preset
Initial amplification is set to zero, and final is set to 100 for a fading in effect.

Fade Out Preset
Initial amplification is set to 100, and final is set to zero for a fading out effect.

Pan L->R, Pan R->L Preset
Initial and final values are set so that the sound starts at one channel and pans to the other.

Echo... echo... o...

Create echoing and reverb effects.   This function creates a continuous echo.   Each successive echo decays in amplitude by the falloff ratio.   To create the effect of a single echo, use the Delay function instead.

Falloff Ratio

Each successive echo will be a certain percentage less than the previous one.   Choosing a falloff ratio of zero would result in no echo at all, while choosing a ratio of 100 would produce an echo which never gets quieter.

Delay

This is the number of milliseconds to place between each echo.   A delay of 100 milliseconds is equivalent to a 1/10th of a second pause between echoes.   Choosing very small values of delay produces quite interesting effects.

Initial echo volume

This is the volume at which the echoes will be mixed with the original sample.   Choosing smaller percentages (30% or so) is nice if the effects of the echoing at 100% make the sound incomprehensible.

Continue beyond selection

Choosing to continue beyond selection will echo the highlighted selection over the rest of the unhighlighted area, stopping at the right-hand edge of the wave that can be seen in the window.   If the window is zoomed in, the echoing will stop before the end of the file, since it will stop at the right hand side of the portion on screen.   By using this option a single word, for example, can be highlighted and echoed over other audio, without echoing the other audio as well.

Echo Left to Right

Selecting this option will make the echoes travel back and forth between the left and right channels.

If you wish to echo the right channel only, select an initial echo volume of 100% for the right, and 0% for the left.

Equalizer

The echo "quick filter" lets you choose approximately which frequencies get removed from the echo first. A setting of zero will leave the frequency band unchanged.   You can choose the frequencies that are "absorbed" as the echo progresses.   The echoed sample is re-filtered through the quick filter on each successive echo.   Setting all values to zero turns off the equalization, since no frequencies are to be absorbed.

This function supports Presets.

Flange

What *is* flanging you ask?   Just try it out and see!   The term is coined from the flanging mechanism on the old style tape recorders which, when fiddled with, would slow down the playing of the tape, and speed it back up again when desired.   That is how they got those funky psychedelic sounding recordings in the 60's.   Here's how you can do it today.

Original - Delayed slide
This slide decides at what proportions to mix the original and flanged signal.   If the Original is at 100%, no flanging is heard.   If the Delayed is at 100%, a cute wavering (like a bad tape player) sound is heard. Portions of both signals need to be present for there to be cancelling out, and reinforcing of wave patterns between the two signals.

Initial Delay
Flanging will start with the delayed signal this many milliseconds behind the original.

Final Delay
Flanging will end with the delayed signal this many milliseconds behind the original.   If the delays are the same, the effect disappears, since the delayed signal will not change.

Stereo Phasing
The right channel can be at a separate delay than the left channel.   A phasing of 180 will put the right channel at the initial delay value when the left channel is at the final delay value, and vica versa.

Rate settings
The **Frequency**, **Period**, and **Cycles** settings are all interrelated, and refer to the rate at which the delay cycles between the initial delay and the final delay.   The flanging will cycle *frequency* times per second, or *period* seconds per complete cycle, or a total of *cycles* complete cycles over the entire selection. Various effects can be heard by using different settings.   For example, if 0.5 cycles is chosen, the selection will start with the initial delay, and end with the final delay.   If a frequency of 4 is chosen, the flanging will cycle from the initial delay to the final delay and back again 4 times per second.

This function supports Presets.

Special Efx

Special Efx is similar to flanging, except that when the delay is at zero milliseconds, The signal is cancelled out to silence, instead of reinforced to full volume as in flanging.   This also creates cool 60's sounding effects.

Initial Delay
Signals that are this distance from the original will be inversely mixed with the original.   The delay will range from this initial delay to the final delay, and back again in a linear fashion.

Final Delay
This is the final delay value.   Please experiment with various settings.

Rate settings
The **Frequency**, **Period**, and **Cycles** settings are all interrelated, and refer to the rate at which the delay cycles between the initial delay and the final delay.   The special efx will cycle *frequency* times per second, or *period* seconds per complete cycle, or a total of *cycles* complete cycles over the entire selection.   Various effects can be heard by using different settings.   For example, if 0.5 cycles is chosen, the selection will start with the initial delay, and end with the final delay.   If a frequency of 4 is chosen, the special efx will cycle from the initial delay to the final delay and back again 4 times per second.

This function supports Presets.

Wave (Brainwave Synchronizer)

Audio may be Waved to produce files, that when listened to with stereo headphones, will put the listener into any desired state of awareness.   For example, by listening to waved files, you can easily achieve states such as deep sleep, theta meditation, or alpha relaxation.

Highest Frequency
This is the brainwave frequency that is represented by choosing points at the top of the graph.

Lowest Frequency
This is the brainwave frequency that well be encoded when points are chosen at the bottom of the graph.

Frequency Graph
Click on the graph to add new control points.   Drag a control point up or down, or off the screen to remove. Choose the highest and lowest frequencies that are represented on the graph with the scroll bars.   Gliding about 4 to 5 Hz over 2 minutes works nicely.   If large variations are done in short time spans, the effects are not as pronounced.   For example, after 5 minutes of Theta waves, if 30 seconds of alpha waves are generated, and returned to theta, the listener will become slightly awake, and aware of his surroundings for that brief moment.   The effect is like all of the sudden changing gears, and you stop thinking about whatever it was you were thinking about, and become aware that you were thinking about it, but aren't any more.

Intensity
This is the intensity of the brainwave encoding.   Higher intensities work well with lower brainwave frequencies.   Beta waves should have intensities below 25 or so, while Delta waves work better with intensities above 60.

Centering
You may choose to have your brain think the synchronization frequencies are coming from the left or right.   This may affect the left or right hemispheres more intensely, but that's only a guess.   Mixing a file that has been waved to the left with one that has been waved to the right (in the same frequency range within 2 Hz) has interesting effects.

Musical Source
If the selection being waved is musical, checking this will calculate the wave patterns in such a way as to eliminate clicks and pops.   If the source is noisy (waterfall, ocean, nature recordings, etc.) do NOT check this.   If you do, it will actually add interference. Since noise is based on "randomness", the clicks and pops are inaudible.

Please read on for more information about Brainwave Synchronization Files

Channel Mixer / Inverse

On stereo waveforms, the channel mixer allows one to have total control over the left and right channels. The default values will leave the wave unchanged.   For mono waveforms, the wave is inverted (that is, crests become valleys, and valleys become crests).

New Left Channel
The slide bars give the percentage of each channel, left and right, that will go into the final wave after mixing.   Choosing an L of 0, and an R of 100 will make the left channel equal to the right channel.

New Right Channel
These two slide bars do the same, but for the right channel.

Invert
Choosing invert for either channel will invert the channel.   Peaks become valleys, and valleys become peaks.   By inverting *both* channels, there will be no perceived difference in sound when listened to.   But, inverting only one channel will greatly change the sound when listened to.

This function supports Presets.

Vocal Cut Preset
This will sum the left channel with the inverse of the right, and place the result into both channels.   On music where the vocals are heard equally loud on both channels, the vocals will disappear, or come close to disappearing.

By playing with the combinations, effects of swapping channels, creating a mono sounding wave that is equal to the left, right, or a mixture of both channels, and creating waves whose left channel is the inverse of the right can be done.   Waving inverse noise has some interesting effects on the mind.

Noise Generator

Generate random noise in a variety of colors.   Each color has its own characteristics. One use for generating noise is to create a waterfall-like sound which is ideal for Waving.   It is also great for making weird effects by flanging and equalizing.

Color
Noise can be a variety of colors, which describe its spectral composition.

> **Brown** noise has a spectral frequency of 1/f^2.   Which means, in English, that there is much more low-end, low-frequency components to the noise, which results in thunder and waterfall like sounds.   Brown noise is called that because, when viewed, the wave follows a Brownian motion curve.   That is, the next sample in the waveform is equal to the previous sample, plus a small random amount.   This gives the appearance of a mountain range when graphed.   The wave pattern is very predictable.

> **Pink** noise has a spectral frequency of 1/f and is found mostly in nature.   It is the most natural sounding of the noises.   By equalizing, rainfall, waterfalls, wind, rushing river, and other natural sounds can be generated.   Pink noise is exactly between brown and white noise (which is why some people used to call it tan noise, but pink was more appealing).   It is neither random, nor predictable.   It has a fractal like nature when viewed.   When zoomed in, the pattern looks identical to when zoomed out, except at a lower amplitude.

> **White** noise has a spectral frequency of 1.   In other words, equal proportions of all frequencies are present.   Because the human ear is more succeptable to high frequencies, it sounds very "hissy".   White noise is generated by choosing random values for each sample.

Style
Noise can be generated in a variety of styles for your listening pleasure.

> **Spatial Stereo** noise is noise generated by using 3 unique noise sources, and spatially encoding them to appear as if one is coming from the left, the other from the center, and the last from the right.   When listened to with stereo headphones, the mind perceives sound coming from all around, not just in the center.

> **Independent Channels** noise is generated by using 2 unique noise sources, one for each channel.   The left channel's noise is completely independent of the right channel's noise.

> **Mono** noise is generated by using 1 noise source, with the left and right channels set equal to the same noise source.

> **Inverse** noise is generated by using 1 noise source as well, but this time with the left channel's noise exactly inverse of the right channel's noise.   When listened to with stereo headphones, the effect is that of the sound coming from the center of the listener's head instead of out in space somewhere.

Intensity
With higher intensities, the noise becomes more erratic, and sounds harsher and louder.

Duration
This is the number of seconds of noise to generate.   If long periods of noise are desired, it is faster to generate a short period of noise (about 10 to 20 seconds), delete excess noise at the beginning and ending of the noise so that the waves are starting and ending at the midpoint, copy, then loop paste as many times as needed.

If a selection range is highlighted, it is **not** replaced by the noise generated.   Noise gets inserted at the insertion point represented by the yellow arrows.

Cool Scripts

Scripts are similiar to Macros.   Your exact mouse moves, and tweaking of parameters is not stored, only the final result when you click "OK".   Undoing something is recorded into the script, since you may have copied something, then undid it, which can be a very useful technique.

Multiple scripts can be kept in one script file, and identified by name.   At any time, you can edit the script file directly to take out steps, rename scripts, remove unwanted scripts, etc.

There are various types of scripts, which depend on when you initiated the recording:
- Scripts that start with File->New, and always start with a blank, empty waveform.
- Scripts that start when a waveform is opened, and work at the current sample rate, etc.   Actions begin at the insertion point in the waveform, and may affect any part of the entire wave if present.
- Scripts that start with a highlighted waveform portion.   All actions in the script pertain only to the portion that is highlighted, leaving the rest of the waveform untouched.

Scripts that run during all of the above conditions will be displayed, but only the ones recorded under the same circumstances will be allowed to run.   In other words, if a script recording started when a portion of a wave was highlighted, then you will only be able to run that Script when something is highlighted.

Scripts are very useful for remembering how you generated a particular sound effect.   Use the script to reproduce the sound effect without having to save the entire waveform.   This is especially useful when generating large brainwave "theta" files, which can take monsterous amounts of space.   By generating the file once, with the scripting turned on (record), you can generate the file again at any time in the future, and save all that hard drive space.   You can also pass along scripts to your friends across email or BBS systems, since they take nearly no memory to store.

When running a script, you can either stop at each dialog box, or have the script automatically run through completion by using the "Stop at Dialogs" checkbox.   Stopping at each dialog box is handy if you wish to 'tweak' the parameters while the script is running.

After recording a script, you may enter a description at the bottom of the dialog to go with the script you just recorded.   This description will appear when the user of the script highlightes the script to run.   Note: the only time you can edit the description is after recording, not before, and not after it has been added to a script collection file.   But, you can still edit the description at any time by pressing the **Edit** button to edit the text file directly.

Important Note: Other buttons and functions are not disabled while the script is running.   Therefore, do not use the other functions until the script has stopped playing.

Customizable Toolbar

If you wish to re-arrange the toolbar buttons, or make certain buttons visible when the toolbar is shorter, you can do this by modifying your cool.ini file.   Cool.ini is located in your Windows directory.   In a section of the INI called [Toolbar] you can define button order, and priority (which buttons get displayed first if there is not enough room for all buttons on the toolbar).   I appologize for there not being a visual way of re-arranging buttons at this point, but it will be possible in the near future.   Until then, you must edit cool.ini.

The easiest way to quickly get a button that you desire on the toolbar *when* you want it is to:

Swap the priority ratings of the button you want to see with a button you do *not* want to see.   For example, if the Paste button is more important than the Loop button, and Paste has priority 37, while Loop has priority 10, then swap the priorities by giving Paste a priority of 10 and Loop a priority of 37.   This way, the Paste button will be displayed before the Loop button as the toolbar grows in length by resizing the application.

Positioning buttons is accomplished by changing the order in the list.   You will notice that each button has an Item and Priority key associated with it.   For example, the Loop button has Item33=Loop and Priority33=10.   If you wanted to move this button to the left, give it a lower key value, such as Item9=Loop and Priority9=10.   When you do this, make sure that the original button associated with Item9 is changed as well, since there cannot be two keys of the same name.   The easiest thing to do is just to swap, and give the Item9 and Priority9 values to Item33 and Priority33.   For example, if Item9=Copy and Priority9=36, then you would change it to read Item33=Copy and Item33=36.   You could also bump up all key numbers by one until you bump Item32 to Item 33.   (ie Item32... becomes Item33..., Item31... becomes Item30... ,etc.).   Again, in the future, all this will be automatic and done visually by dragging and dropping the buttons.

If your toolbar becomes messed up, and you want it back to the original, just delete the entire [Toolbar] section along with all Item and Priority items, and Cool will re-build the section with the defaults.

About the Author

That's me, David Johnston.   I have been working with various sound programs since about 1991, and have been toying with synthesizers since the 70's.   I have tried to include all the wave editing features I have found useful.   I welcome any suggestions for new waveform transformation functions.   The "wave" function in this program is quite unique, and is the same function as in the freeware program out called MINDSYNC (wave.exe/wave87.exe); see <u>Wave</u> for more details.   This program should be compatible with every sound card on the market that has Windows drivers available for it.   Please read the special notes in the Contents section If you have problems.

If you find this program useful feel free to let me know!

The Demonstration version of Cool has the following restrictions:
- You have unlimited use of any of the functions, but may only use 2 groups of functions per editing session.

If you have the demo version, and want more information, please write:

> David Johnston
> 28022 NE 147th Place
> Duvall, WA   98019

I am considering US $20 - $25 a fair registration fee.   If you register, you will be sent a registration number that will *unlock* the software, allowing unlimited use of all functions all the time.   Keep in mind that even though this product is Share-Ware now, it may not always be that way, and you may be asked to purchase the full retail product instead (that is, if there ever *is* a full retail product.)

Demonstration Copy

Currently, you have the demonstration version of Cool Edit, which has 100% functionality on 2 function groups per editing session.   If you wish to use another function that is currently disabled, you must exit all instances of Cool-Edit, and re-start it to choose the functions you desire.

If you would like more information on the program, please write:

>   David Johnston
>   28022 NE 147th Place
>   Duvall, WA   98019

I am considering US $20 - $25 a fair registration fee, although if you think it is worth more, then I will gladly accept more :).   If you register, you will be sent a registration number that will *unlock* the software, allowing unlimited use of all functions all the time.   Keep in mind that even though this product is Share-Ware now, it may not always be that way, and you may be asked to purchase the full retail product instead (that is, if there ever *is* a full retail product.)

You may include your internet address for a faster response.

New Wave

When creating a new waveform, you must specify the waveform properties.   using higher sampling rates, stereo, or higher bit resolutions will result in higher quality sounds at the expense of requiring more memory.

Sample Rate
This describes how many times per second to take a *snapshot* of the audio.   The human ear can perceive sounds up to about 17,000 cycles per second, or 17 Khz.   When choosing a sample rate, frequencies of up to 1/2 the sample rate can be produced effectively.   So to reproduce frequencies up to 10Khz, a sample rate of at least 20Khz must be chosen.   Choose **Custom** to enter a sample rate not listed.

11,025 Hz	Poor AM Radio Quality
22,050 Hz	Near FM Radio Quality
32,075 Hz	Better than FM Radio Quality
44,100 Hz	CD Quality
48,000 Hz	DAT Quality

Channels
Mono waveforms support one channel of audio information.   Stereo files take twice the space because there are two channels of information represented, a left and a right channel.

Resolution
This describes the number of bits to use for each sample on each channel.   Choosing 8-bit resolution will provide 256 unique "volumes".   The PC-Speaker, for example, provides only 4-bits of resolution because it can support 16 unique volume levels.   Choosing 16-bit resolution will provide 65,536 unique "volumes", for a 96 dB signal-to-noise ratio.   Much quieter sounds can be reproduced at 16-bit resolution than at 8-bit resolution, which only has a 48 dB signal-to-noise ratio.   Compact disk players have a 16-bit resolution.

Note: Certain combinations of sample rate, channels, and resolution may not be available on your system.   To see the maximum capabilities of your system, look at the status window when starting the program.   you may create and edit files that cannot be played on your system, you just may not be able to play them properly.

Adjust Sample Rate

You can change how Cool-Edit interprets the actual waveform data by adjusting any of the parameters below associated with the wave.   Be aware that listening to Mono data in Stereo format, or listening to 8-bit data in 16-bit format will sound really really weird!

Adjusting the various parameters comes in very handy when loading in waveforms of unknown type (RAW).   You can play with the various settings until the wave sounds "right".

Sample Rate
This describes how many times per second to take a *snapshot* of the audio.   The human ear can perceive sounds up to about 17,000 cycles per second, or 17 Khz.   When choosing a sample rate, frequencies of up to 1/2 the sample rate can be produced effectively.   So to reproduce frequencies up to 10Khz, a sample rate of at least 20Khz must be chosen.   Choose **Custom** to enter a sample rate not listed.

Channels
Mono waveforms support one channel of audio information.   Stereo files take twice the space because there are two channels of information represented, a left and a right channel.

Resolution
This describes the number of bits to use for each sample on each channel.   Choosing 8-bit resolution will provide 256 unique "volumes".   The PC-Speaker, for example, provides only 4-bits of resolution because it can support 16 unique volume levels.   Choosing 16-bit resolution will provide 65,536 unique "volumes", for a much higher signal-to-noise ratio.   Much quieter sounds can be reproduced at 16-bit resolution than at 8-bit resolution.   Compact disk players have a 16-bit resolution.

If you wish to convert your current sample to a new sample rate, please read about Converting Sample Rates for the correct procedure.

Note: Certain combinations of sample rate, channels, and resolution may not be available on your system.   To see the maximum capabilities of your system, look at the status window when starting the program.   you may create and edit files that cannot be played on your system, you just may not be able to play them properly.

Byte Swapping

You can change how Cool-Edit interprets the actual waveform data by swapping high and low order bytes in 16-bit mode, or by adjusting the signed/unsigned interpretation of bytes in 8-bit mode.   Swapping the byte ordering comes in very handy when loading in waveforms of unknown type (raw PCM).   You can try swapping the bytes if the waveform does not sound correct.

Paste Special

Waves from the clipboard can be looped, or mixed the current waves.   They are inserted or overlapped starting at the current <u>insertion point</u>.

Volume
Use the volume slides to paste an amplified version of the clipboard wave into the current waveform.   By adjusting the volume slides, single channels may be pasted.

Lock left/Right
When checked, the volume slide bars are locked, so both left and right volumes can be adjusted at the same time.

Overlap
When overlap is checked, the clipboard wave does **not** replace the currently highlighted selection, but is mixed at the selected volume with the current wave.   If the clipboard wave is longer than the amount selected, the wave continues being pasted beyond the selection.

Loop Paste
When checked, the clipboard wave is pasted the number of times entered.

If the format of the waveform data in the clipboard differs from the format it is being pasted into, it will be converted accordingly before pasting occurs.

From Clipboard
When chosen, the audio data to be pasted is the data currently on the clipboard.

From File
When chosen, a file may be chosen with **choose file** to be pasted.   This is especially useful when the amount of data you wish to paste is too large for the clipboard.   Simply save the data you want to mix to a file, and paste in that file.

About Carrier Waves

A carrier wave is needed to transport the brainwave frequencies.   Because the carrier wave is not what you hear through the headphones directly, you do not need to buy super high-end headphones (5Hz-25KHz) to reproduce the effects.   In other words, your headphones do not need to be able to reproduce a 5Hz signal if you are generating a 5Hz theta-frequency brainwave file.   The brain *does* however respond better to the lower frequencies, so the better the headphones you buy, the more dramatic the results will be.   The best headphones are the kind that cover the entire ear, so outside noise does not get in.   Plus, these headphones have much higher response to low frequencies.

Carrier waves must have some correlation between the left and right channels, no matter how slight.   So mono (total correlation), inverse (total negative correlation), and spatial (natural recordings that have some of the same sounds coming in both channels) will work great.

The best sounds to use as carriers are sounds that are spread across the entire frequency range, or at least most of the lower frequency range.   Good examples are ocean, waterfall (most any recordings from nature), and noise generated by this program.   Experiment with mono (both left and right channels the same), inverted (like mono, but the left channel is the inverse of the right, obtained by using the Channel Mixer), and spatial stereo (spatially encoded sounds in nature, recorded with microphones about 9 inches apart to simulate separation between the ears).   But don't let this stop you from digitizing your favorite music, and using it as a carrier, or converting your favorite to a mono or inverted wave.

To generate a carrier wave, you can do three things:

Record a sample   Once recorded, use the Channel Mixer to create a mono, or inversed wave.   Or just leave it the way it was recorded.   You may find changes in effectiveness of the brainwave files depending on how you use the Channel Mixer.   Keep in mind that this function only operates on stereo waves, so when "mono" is mentioned, it means that the exact same signal is present on both channels--the left channel and right channel are the same.

Generate Tones   You may use the Generate Tones function to find a pleasing, relaxing tone for the background (but I find "noise" sounds more relaxing).   The way tones work the best is if the left channel's tone frequency is 5-6 Hz different from the right channel's tone.   This creates a beat pattern equal to the frequency difference, which the brain responds to somewhat (this is the property that many theta-inducers rely on).   To do this, generate one tone with left volume at 40, and right volume at zero.   Then generate the second tone with the left and right volumes reversed.   Finally, Paste Special (with overlap) one tone on top of the other.   Use low frequency tones, like 50Hz to 120Hz for best results.   These tones, by themselves, will help coerce the mind into the state associated with the difference between the frequencies.   For example, for a theta state of 6Hz, use a 70Hz and a 76Hz tone.   Combining this tones sample with an existing brainwave file, by overlap pasting at a quiet volume (20%) is even more effective.

Generate Noise   Use the Generate Noise function (pink and brown work best) in any of the modes: mono, inverse, or spatial stereo (independent channels noise will **not** work as a carrier for brainwave frequencies at all, since there is no correlation between the left and right channels).   I find that using pink noise in spatial stereo, and running it through the Quick Filter to get rid off some of the "edge" if any works the best.   I have also found Inverse to work quite well too, but the brainwave "effect" is more pronounced, and can be distracting, and some sound boards have trouble reproducing sound that is inversed between channels.

Once you have found a pleasing sound, about 10 seconds or so of a monotonous sound (tones, river, waterfall, noise...) you're ready to start.   If a monotonous sound is used, more disk space can be saved because we will use the play list to repeat portions.   If a music sample were used, it is quite noticeable that the same 10-second piece is being played over and over and over again.

If you're curious you can also <u>spatially locate a mono sound to the left or right?</u>.   Do this if you wish to

have the illusion that a particular sound is coming from one side or the other.   The function works by pasting a mono sound sample into a stereo waveform, and using the Digital Delay function.   Having a quiet "ping" (generated by using the sine wave tone generator with the bell curve envelope) play spatially on the left, then on the right at about 5 second intervals is very relaxing.

Encoding Brainwave Information

There are two types of brainwave files that you can create: A **flat file**, and a **cued file**.   The flat file takes more memory, and plays straight through from beginning to end, while the cued file is actually contains pieces of the entire audio program, that when played in the proper order become the brainwave file.   The cued file takes less memory, and can very quickly be modified at any time by re-arranging the audio pieces.   The average length of a cued file is about 3-4 minutes for a program that can last as long as desired.   The flat file is a standard wave file, which means to create a long program, you must have enough space for it.   The only advantage to using a flat file is if you are waving music, since music cannot be split into pieces and re-arranged, otherwise it would sound discontinuous.   Creating brainwave files using the flat file method will be discussed first, since it is more straightforward

Flat Brainwave File Generation   Create a file the length you wish to make your relaxation program using the carrier wave(s) of your choice.   Either record music, or use the pink noise generator and copy and paste (or Paste Special) to the desired length.   If you are using a monotonous sound, you would be better off using the cued file method.   Lengths of good relaxation programs vary from 15 to 30 minutes, and beyond.   This means you must have enough hard drive space for the entire file.   Since the temporary file takes up hard drive space as well, the maximum size of file you can create, and be able to save, will be one that takes up half of the initial free hard drive space.

Use the Wave function to encode the brainwave patterns into the carrier wave by highlighting a section of the wave, or the whole thing, and choosing Transform:Wave, or click the wave icon.   With the wave transformation, you have complete control over the brainwave frequency being encoded, the strength of the signal, and the positioning of the signal left or right.   Over the selection highlighted, the intensity, and position remain constant, but the frequency can be varied using the graphical input control.   See the section on Authoring Brainwave Files to learn what settings to use for the Wave function, and how to build effective files.

Once the entire file has been waved to your satisfaction, you can save the file if you wish, and play it using the Play button.   An interesting side effect is that different sounds are heard if you listen to one channel, listen to both channels with one ear, or listen to each channel with each ear.

Cued Brainwave File Generation   These files contain many short snippets of brainwave encodings at different frequencies.   Each snippet is cued using the Cue List, and a Play List is generated by adding entries from the Cue List, and looping them if necessary.   To listen to a cued brainwave file, you must use the Play button in the Play List dialog box.

First you must figure out how you want to divide up the brainwave program (your 20-30 minute masterpiece) into components.   For example, you may want to have patterns of 5Hz, 7Hz, and 9Hz at different points in the program.   In this case, you will need at least three pieces for your creation.   The actual file will just be 10 seconds of carrier wave at 5Hz, followed by 10 seconds at 7hz, followed by 10 seconds at 9Hz.   All the pieces are placed in the cue list by highlighting the piece, and choosing **Add**.   It is best to add the piece to the cue list once it is created, or pasted at the end of the current waveform.   To create the final program, the pieces are added to the Play List in the order you wish to listen to them. Each piece can be looped if needed.   So a 20 minute program can be generated from 3 10-second pieces by adding the cues to the play list and looping.

First you need to create 10 to 20 seconds of carrier wave, and save in a special file in case you need the carrier wave again later.   Highlight the wave, and Edit:Copy.   When you need another copy of the initial carrier wave, you need only to Paste it.

Add the first carrier wave snippet to the Cue List by pressing the **Add** button in the Cue List dialog.   Give the cue for this snippet a name that reflects the waveform transformation you will be using, for example, "6Hz to 5Hz drop".

Choose the **Transform:Wave** function to encode the proper patterns into the carrier wave.   Look at the section on Authoring Brainwave Files to learn what settings to choose.

Click past the end of the wave file (make sure the rightmost part of the file is in view), and choose Paste to insert another copy of the carrier wave.   Once you do this, you can add the newly inserted selection to the cue list, and give it a name.   Repeat the step above for creating a brainwave encoding over the carrier wave you just inserted.   Do this as many times as needed until you have all the pieces you need to build the final brainwave file.

Once all the pieces have been generated, add them in the order you like to the play list.   To make pieces last longer (if the beginning and ending of the piece are at the same brainwave frequency), increase the number of loops for that entry in the play list.

When Played from the play list, the pieces will be played in the order shown, and looped if necessary.

To get familiar with the cue list, and play list, open one of your favorite wave files, and highlight sections then add them to the cue list.   After you have a few selections in the cue list, add them to the play list, and choose a loop count of greater than one for some of them.   Choose Play from the play list, and listen to what you've just created.

Authoring Brainwave Files

After learning about carrier waves, and encoding procedures, all you need to know is what frequencies to use, and when to use them during the course of the listening session.   Once you know what frequencies to use, and at what intensity, you can generate the completed file using either of the methods above.

Effective brainwave files have some sort of encoding going on the entire length of the session.   For the first 3 minutes or so of the session, the listener will not be in a "relaxed" state, and will not respond greatly to the frequencies being presented.   During this *warm-up* period, gradually decreasing from about 12Hz down to 8Hz works nicely.   After about 4 minutes, the listener's brainwave patterns will start to synchronize with the patterns in the headphones, and the serious brainwave programming can begin.

Frequencies of 8-10Hz correspond to an alpha state -- light relaxation, like a quick afternoon siesta. Frequencies of 6-7Hz correspond to a theta state -- meditation.   4-5Hz correspond to deep relaxation. You can create a session that is constant, in one of these states, or create a session that dynamically flows from one to the other.   When going down in frequency, give the listener about one minute to "catch up", and stay in sync with the wave.   Going up in frequency does not require the listener to catch up.   In other words, if you go from 6Hz down to 4Hz over a 20 second timespan, and hold at 4Hz, the listener may not be at 4Hz for another minute.   When going from 4Hz to 8Hz in 20 seconds, the listener will be at 8Hz at the end of the 20 seconds.   It appears to take extra time when going down in frequency, but no extra time when going up.   This basically holds true for the first 20-30 minutes of a session.   After that, the opposite tends to occur.   It is easier to go lower than go higher.   This means that to bring a listener from 4Hz (where she has been for the last 30 minutes) up to 12 Hz, it should be done over a 5 minute period or so.   One nice "trick" to do is to keep the listener at around 4-5 Hz for a while, then about once every 2 minutes, go up to 8Hz and back over a 20 second span.   This will *alert* the listener slightly, and make them aware for a few seconds of what they are thinking.   This is great for getting creative insights and the like.   It acts as a sort of *window* to the subconscious, allowing one to remember what is going on. It's kind of like remembering dreams: you do it better if you are awaken in the middle of one.

Another effective method of producing relaxation files is to overlap them.   That is, have portions that are one frequency, and slightly spatially located to one side overlapped with a slightly differing frequency spatially located slightly to the other side.   This gives the listener the chance to *decide* which frequency to be at, and gives them more freedom over the experience.   For example, a session could go from 8Hz to 4Hz over 10 minutes overlapped with 7Hz to 5Hz over the same 10 minutes.

For nice *super-relaxing* effects, generate panning waves (frequencies of 0.05 to 0.2) over your session after encoding the initial brainwave patterns.   For example, if you are generating a brainwave file out of 20-second pieces, after generating the main brainwave frequency over the 20 second period, generate a panning wave of 0.05 or 0.1 (which means a period of 20 or 10 seconds) with an intensity of about 50 or so.   This will make the sound appear to shift left and right to the listener over a 20 or 10 second period. Now, overlapping a 24-second piece panned at 0.125 (8 second period) at 5Hz with a 0.167 (6 second period) at 6Hz will combine the practices of multiple frequencies with panning for an extremely super-natural effect!

Once you get started creating a few files, and see what the different frequency ranges do, you will become familiar with the different effects and how to generate just the effects you want.

Sample Theta File - Step-by-Step

1 Create a new blank file with **File:New**.   Choose a **Stereo** file, either 8 or 16 bit and a 11025, 22050, or 44100 sampling rate.   The final file size will be one of the following sizes listed below depending on your choice:

	11025	**22050**	**44100**
8-bit	2.6M	5.2M	10.5M
16-bit	5.2M	10.5M	21.2M

You must make sure you have enough memory for a file of this size, plus an additional meg for working space.   If you plan on saving the file when you are done, you must have at least **twice** this amount of hard drive space available, since a temporary file is used instead of memory while working on the wave.

2 Choose **Generate:Noise**.   Choose **Pink Spatial Stereo** for **15** seconds at an intensity of **3**.   This is usually the longest portion of the generation of brainwave files.   Because of this, it is advised that you save this piece of *noise* so that in generating future files, you can just load in this pre-calculated noise as a starting point.

3 Choose **Edit:Copy**.   From now on, we will paste the noise in when we need it!

4 Make sure the noise is highlighted.   If it is not, select all by double-clicking on the waveform until it is highlighted.

5 Choose **Add** in the Cue list, and give the entry a **Label** of **10Hz to 8Hz**, and a **Description** of **Warm-Up**.

6 Choose **Transform:Wave** to bring up the brainwave dialog box.   Enter **10** for the **Highest Frequency**, and **8** for the **Lowest Frequency**, and an **Intensity** of **35**. On the graph above, click the leftmost dot, and drag it to the top of the graph.   Click the rightmost dot, and drag it to the bottom of the graph.   This will product a frequency encoded at 10Hz at the beginning, and glide down to 8Hz by the end.   Choose **OK** to generate the encoding.   This shouldn't take nearly as long as it did to generate the noise.

7 Click the mouse at the rightmost portion of the wave (just beyond the *black* waveform display area). When you do this, the yellow cursor arrows should be all the way to the right of the wave. You must always add new pattern blocks at the **end** of the current waveform.

8 Choose **Edit:Paste** to insert another copy of the original noise that we had copied originally.

9 Create the following pattern blocks as before (following the steps 5 to 8) , except with the following values for the cue list and waveform transformation:

Label	Description	Hi Freq.	Lo Freq.	Intensity
8 Hz	Alpha	8	8	37
8 to 6Hz	Glide Down	8	6	38
6Hz	High Theta	6	6	40
6 to 5Hz	Deeper Theta	6	5	45
5Hz	Theta	5	5	50
5-8-5	Spike	8	5	50

(Graph should look like an upside-down "V")

Label	Description	Hi Freq.	Lo Freq.	Intensity
5 to 12Hz	Awake	12	5	40

(Graph should go from left=lowest to right=highest)

10 Once all the blocks are generated, and in the cue list, Add the pieces to the play list by selecting the wave portion in the cue list and clicking **Add** in the play list. Select the pieces listed below in the order given.   After doing so, select each item in the play list, and change the **Loops** for each so the final play list looks like this:

(1)	10 to 8Hz
(3)	8Hz
(1)	8 to 6Hz
(7)	6Hz
(1)	6 to 5Hz
(18)	5Hz
(1)	5-8-5
(12)	5Hz
(1)	5-8-5
(12)	5Hz
(1)	5-8-5
(12)	5Hz
(1)	5-8-5
(12)	5Hz
(1)	5 to 12Hz

When you choose **-Play-** from the play list, the sequence will be played in the order given, looping the number of times specified.   This list gives a 21 minute theta session, with bursts into alpha at four points.

11 If you wish to save this piece, and have enough hard drive space, you can do it now.   The wave is complete.   Enjoy.

About Brainwave Files

The wave option works like many meditation tapes and light/sound devices on the market, which range in price from $200 to $500.   There's even a board available with plug in glasses (which have blinking lights) for your PC for $495.   I think the files created using the 'Wave' transformation are even more powerful, and are definitely more pleasing to the ears.   Most other devices and tapes have a "humming" sound or some other tones to induce the right brainwave frequencies.   This program allows you to use ANY sound to encode the frequencies with.   The most effective I have found are by using the noise generator, which creates pleasing waterfall like sounds.

There are four major Brainwave Patterns ranging from deep sleep to acute awareness.   Listening to sounds that have been waved for periods of 5 minutes or more will produce the desired state of awareness in the listener.   Sessions of 25 minutes or so work really well!

Many Uses

About Carrier Waves
Encoding Brainwave Information
Authoring Brainwave Files
Step-By-Step Theta File
How to Use Brainwave Synchronization Files

Articles on Theta meditation

DISCLAIMER
By using this program, you agree that the author of the program will not be responsible for any damage as a result, direct or indirect, of using this program.

WARNING
Sounds generated by the wave function are not suitable for epileptics or persons undergoing psychiatric treatment since it *does* alter ones state of consciousness.

Many Uses

Relaxation
Choose between 5hz and 10 hz for different levels of relaxation.

Meditation
Choose between 4hz and 7hz, either cycle between a few, or stay at a particular frequency for different results.

Induce Sleep
Choose between 4hz and 6hz for starters, then go into frequencies below 3.5hz, settling on about 1.5hz to 2.5hz for sleep.

Creative Visualization
About 6hz for a while, then up to 10hz works well.

Stress Reduction
Any use of frequencies below 11hz will reduce stress.

Self Hypnosis
Choose about 8hz to 10hz while playing any self-hypnosis tape, or guided meditation.

Super Learning
Choose about 7hz to 9hz while playing any learning tapes, like foreign language tapes, etc. to increase comprehension.

Subliminal Programming
Choose 5hz to 7hz while playing your favorite subliminal tapes, or make your own by recording some affirmations, and mix paste them from the clipboard at barely audible volumes.

Improve ESP / Intuition
Theta frequencies help in this area, 4hz to 7hz.

Reaching Higher States of Consciousness
Theta again, with daily half hour sessions.

Quick Refresher on long days
Low Alpha 8hz to 10hz for about 15 minutes works well.

How To Use

Once you have created your brainwave file (15 minute files on up work best), <u>loop play</u> them for a longer listening time.   Sessions of 15 minutes or more work best.   It is best to listen to the sessions lying down in a quiet place where you will not be disturbed.   If there is no place like this near your PC, it may be a good idea to record the session on tape and listen to it where you can be comfortable and relaxed. When you're fully comfortable, start the session, close your eyes, and let the magical sounds from Cool Edit do the work.   Remember, this only works if you listen to the   sounds with stereo headphones.

You may notice helicopter, or "washing" type noises moving around in your head.   These sounds are actually created inside your head, and are not coming directly out of either channel from the sound board. It is this noise that is doing the work of helping your brainwaves get synchronized to the patterns you have chosen.   When I have mixed two different (but similar in frequency ranges) brainwave files together, I have noticed a jet airplane noise moving slowly from left to right in the background.   Some people don't hear these artifacts at all, while others hear them extremely well.

Another side effect is that of a wandering mind.   When I use frequencies under 8hz, I find myself thinking of the strangest things.   You may find that you are not thinking of anything in particular, and your thoughts become very interesting.   The feeling is also "warm" and "happy" for some people.   Others start recalling their favorite memories as a child, even some they thought they had forgotten forever!

After a session of 15 minutes or more, you may feel quite refreshed, light, airy, clear-headed, etc.   I always find myself feeling very good afterwards.   Some claim that doing this for 30 minutes a day can result in subtle but great changes in your life.   ESP experiences increase, and you may be able to reach new levels of awareness in your everyday life.

Unsolicited Testimonials
Excerpts from Articles on Theta Meditation

Any results gained from using Light/Sound devices such as the Voyager, the Brain Machine, MasterMind, the MindsEye, etc. can be gained from listening to these files as well. Some of these excerpts are from results using these machines with flickering light goggles and tones.

This is my own testimonial. I find that within minutes of listening to a Theta pattern, my thoughts start drifting to really weird stuff that doesn't make any sense at all. It is really quite funny some of the things I find myself thinking of. Then I start to feel really "good" and relaxed--at peace if you will.

It has been stated in several places that a half hour a day of the brain being in the theta state can replace up to 4 hours of sleep. People find they need less sleep at night.

Thirty patients had sessions in Theta (5 hz) and experienced relaxation states of 80-100% after five minutes as well as improved pain relief. Eight patients had blood tests before and after the sessions and showed improved beta-endorphin levels of 10-50%.

Using a first-generation prototype light/sound device, one doctor noted, "these devices produce a distinct relaxation state. Programming the device between 3 and 7 hz, it takes about 10 to 15 minutes for the patients to enter--effortlessly--a state of hypnosis. They terminate the sessions relaxed and with a feeling of well-being." Also, "the device has a calming effect on the nervous or anxious patients. In a majority of cases, the patients feel relaxed, and calm during a period of three to four days after the session. It happens that the subjects have a reminiscence of childhood experiences, particularly when in Theta. They related their experineces which we incorporated into our psychotherapeutic program."

"The machine works like a tranquilizer and the effect lasts for several days. Using the machines in Theta frequency, clients are very receptive to suggestions on behavioral aspects such as reducing tobacco, alcohol and food consumptions." Many patients "were more creative during the sessions."

"By inducing hemispheric coherence the machine can contribute to improved intellectual functioning of the brain. Like children spending most of their time in Theta, the machine allows a reduction in learning time. With adults a return into Theta allows them to rediscover childhood experiences. The machine is like a 'lost and found office' for the subconscious."

D.J. Anderson, B.Sc., M.B., "The treatment of Migraine with Variable Frequency Photo-Stimulation," in HEADACHE, March 1989, pp 154-155:

> D.J. Anderson used photo-stimulating goggles with variable frequency using red LEDs in order to stimulate the optic nerve, through closed eyes, right and left with frequencies between 0.5 and 50 hz. The study included seven patients who suffered a total of more than 50 migraines during the observation period. Forty-nine of these migraines were relieved (either by reducing the average duration or by increasing the frequency interval in between migraine crisis) and 36 other migraines could be stopped while using the goggles.

The more these sounds are used, the easier it becomes to produce and maintain Alpha/Theta rhythms. As these states of higher awareness become infused into normal brain activity, the result can lead to what some have called a fifth state of consciousness, or an "awakened mind." In this state of illumination and bliss one sees the world as distinctly as before but with a new mind that perceives the universe with new meaning. It's this experience of illumination that is the seed for all breakthrough scientific theories, literary ideas, revolutionary inventions, and artistic masterpieces. The technology used here induces these states by forcing your brain to focus your mental energies inward... tapping your own vast reserve of creative genius and eventually unfolding "an enlightened state of awareness."

An unusual side benefit of listening to these sounds is a surprising need for less sleep. Some users are able to reduce their sleep requirement by as much as 3 - 4 hours each night, rising each morning feeling

refreshed as if they had slept a full 8 hours.   The reason?   It's believed the theta-sounds replace the need for extensive dreaming which is the main purpose of sleep.   Another interesting side effect, many users report a dramatic increase in sex drive.   No one knows exactly why, but it may be linked to changes in brain chemistry.   But, perhaps the most unusual side effect is the reported increase in psychic functioning, including episodes of precognition, out-of-body experiences, and spontaneous channeling events.

When you finish each session your entire body becomes charged with a new energy and vitality.   Fears and anxieties are gone.   You are renewed, more alert, and mentally you feel on top of the world.

What causes the euphoria and peak experiences?   The neuroscientists say the 'high' you experience is caused by a release of endorphins in the brain.   A hundred times more powerful than morphine it makes you feel like you're soaring with eagles.

Zen meditators have been found to alter Alpha/Theta frequency according to their depth of meditation, reports Japan's leading neurophysiologist, Dr. Tomio Hirai.   he has correlated brain-wave patterns with certain stages of meditation.   And according to Dr. Hirai, "Meditation is not merely a state between mental stability and sleep, but a condition in which the mind operates at the optimum.   In this condition the person is relaxed but ready to accept and respond positively to any stimulus that may reach him."

Research now confirms that brainwave rhythms correspond to certain states of consciousness, and this suggests that individuals capable of altering their brainwave patterns can have significant control over other mental and physiological functioning.   As Elmer and Alyce Green of the famous Menninger Institute first reported in the mid-70's, "...simply causing your brain to generate theta activity for a few meinutes each day seems to have enormous benefits, including boosting the immune system, enhancing creativity, and triggering integrative experiences leading to feelings of psychological well-being."

Biofeedback researchers have found that people who enter the "theta state," expand their states of consciousness, acquire super-receptivity to new information, and demonstrate a greater ability to "rescript" material on a subconscious level.   Even more astonishing are the findings of a study conducted on a group of chronic alcoholics at a University in Colorado.   After 13 weeks the group that learned to generate theta and alpha brainwaves, showed a far greater recover rate, and a complete transformation of personality. There is a remarkable body of evidence...

Another Note: I did not do alot of proper citing, I just wanted to present some information that is available "out there".

The major brainwave patterns are:
Beta   > 13 hz Normal state of alertness, stress and anxiety.
Alpha   8-12 hz Light relaxation, "superlearning", positive thinking
Theta   4-7 hz   Deep relaxation, meditation, increased memory and focus
Delta   1-3 hz   Deep sleep, lucid dreaming, increased immune functions

Selecting **Loop/Play Toggle** from the **Options** menu will toggle the play button between the **Loop** and **Play** states.

Selecting **Monitor source** from the **Options** menu will start monitoring the recording source, which is useful for setting the recording levels before recording.   To stop monitoring, press the **Stop** console button.

Use the mouse to click on the starting point of the wave you wish to select, then drag the mouse to the ending point.   When doing this, the portion that is being selected will be inverted (you'll see a white bar indicating the selected portion).   Use the Shift button in combination with the mouse to extend a selection.

Double-clicking in the viewing field will select the entire viewing field.

After selecting a range, **Zoom In** will expand the selection to full viewing area

If zoomed in, zooming out will display more of the wave in the viewing area to the left and to the right.

Clicking on **Full View** will display the entire waveform in the viewing field.

The green bar above the viewing field will shrink or grow depending on whether the view is zoomed in or not.   The location of the green bar depicts the relative location in the waveform the viewing area is displaying.   Clicking on the green bar and dragging left or right will scroll the wave being viewed left or right.   Double-clicking will bring up the samples data entry box.

Beginning and Ending samples are displayed in the boxes at the right side of the window.   Beneath is a time display showing the amount of time selected, or currently being viewed.   Double clicking on any of these displays will toggle the mode between Samples and Time.

Clicking the **Play** button will play the selected range, or the entire portion of the wave in the viewing field if no range is selected.   If the button reads **Loop**, then the selection will be repeated until the **Stop** button is pressed.

Clicking the **Stop** button will stop the currently playing wave from playing.

If your system supports recording, pressing this will start recording.   Press **Stop** when done recording. Recording starts at the insertion point and overwrites any data thereafter if any.

The point of insertion for **Paste** and **Paste Special** is indicated by the yellow arrows above and below the viewing area.   The exact sample is listed in the **Beg:** box.

Filters.h include file

```
//////////////////////////////////////////////////
// GENERAL PURPOSE DEFS FOR CREATING CUSTOM FILTERS //
//////////////////////////////////////////////////
typedef DWORD          FOURCC;          /* a four character code */

struct cue_type
{      DWORD dwName;
       DWORD dwPosition;
       FOURCC fccChunk;
       DWORD dwChunkStart;
       DWORD dwBlockStart;
       DWORD dwSampleOffset;
};

struct play_type
{      DWORD dwName;
       DWORD dwLength;
       DWORD dwLoops;
};

typedef struct coolquery_tag
{      char szName[24];
       char szCopyright[80];

        // rate table, bits are set for modes that can be handled
       WORD Quad32;  // Quads are 3-D encoded
       WORD Quad16;
       WORD Quad8;
       WORD Stereo8;                // rates are from lowest bit:
       WORD Stereo12;               // bit 0 set: 5500 (5512.5)
       WORD Stereo16;               // bit 1 set: 11025 (11K)
       WORD Stereo24;               // bit 2 set: 22050 (22K)
       WORD Stereo32;               // bit 3 set: 32075 (32K, or 32000)
       WORD Mono8;                  // bit 4 set: 44100 (44K)
       WORD Mono12;                 // bit 5 set: 48000 (48K)
       WORD Mono16;                 // bit 6 set: 88200 (88K)   (futureultra-sonic
rates?)
       WORD Mono24;                 // bit 7 set: 96000 (96K)
       WORD Mono32;                 // bit 8 set: 132300 (132K)
                                    // bit 9 set: 176400 (176K)

       DWORD dwFlags;
       char szExt[4];
       long lChunkSize;
} COOLQUERY;

#define R_5500   1
#define R_11025  2
#define R_22050  4
#define R_32075  8
#define R_44100  16
#define R_48000  32
#define R_88200  64
#define R_96000  128
#define R_132300 256
#define R_176400 512


#define C_VALIDLIBRARY 1967


#define QF_RATEADJUSTABLE 1      // if can handle non-standard sample rates
                                 // if not, only rates in bit rate table
```

```
understood
#define QF_CANSAVE 2
#define QF_CANLOAD 4
#define QF_UNDERSTANDSALL 8        // will readANYTHING, so it is the last resort if no
other
                                   // formats match
#define QF_READSPECIALFIRST 16     // read special infobefore trying to read data
#define QF_READSPECIALLAST 32      // read special infoafter reading data
#define QF_WRITESPECIALFIRST 64    // when writing afile, special info is sent to DLL
before
data
#define QF_WRITESPECIALLAST 128    // when writing,special info is sent to DLL after
data
#define QF_HASOPTIONSBOX 256       // set if options boximplemented

// special types are read from and written to DLL in the order below

// special types (particular to Windows waveforms)
#define SP_IART  20
#define SP_ICMT  21
#define SP_ICOP  22
#define SP_ICRD  23
#define SP_IENG  24
#define SP_IGNR  25
#define SP_IKEY  26
#define SP_IMED  27
#define SP_INAM  28
#define SP_ISFT  29
#define SP_ISRC  30
#define SP_ITCH  31
#define SP_ISBJ  32
#define SP_ISRF  33
#define SP_DISP  34
#define SP_CUE   40 // returns number of cues of size cue_type
#define SP_LTXT  41 // returns number of adtl texts of size 8 (4,id and 4,len)
#define SP_NOTE  42 // returns LO=size, HI=number of strings (sz szsz...)
#define SP_LABL  43 // returns LO=size, HI=number of strings(sz sz sz...)
#define SP_PLST  44 // returns number of playlist entries size play_type
```

Myfilter.def Module Definition File

```
LIBRARY   Cool*WAV

EXETYPE   WINDOWS

CODE      PRELOAD MOVEABLE DISCARDABLE
DATA      PRELOAD SINGLE

HEAPSIZE  2048

EXPORTS
            WEP                     @1 RESIDENTNAME
            QueryCoolFilter         @2
            OpenFilterOutput        @3
            CloseFilterOutput       @4
            WriteFilterOutput       @5
            OpenFilterInput         @6
            CloseFilterInput        @7
            ReadFilterInput         @8
            FilterUnderstandsFormat  @9
            FilterCanWriteSpecial   @10
            FilterCanReadSpecial    @11
            FilterWriteSpecial      @12
            FilterReadSpecial       @13
            FilterGetOptions        @14
            FilterOptions           @15
            DIALOGMsgProc           @20
```

Myfilter.mak Sample Make File

```
# Microsoft Visual C++ generated build script - Do not modify

PROJ = WAVE
DEBUG = 0
PROGTYPE = 1
CALLER =
ARGS =
DLLS =
D_RCDEFINES = -d_DEBUG
R_RCDEFINES = -dNDEBUG
ORIGIN = MSVC
ORIGIN_VER = 1.00
PROJPATH = C:\SOURCE2\COOL\WAVE\
USEMFC = 0
CC = cl
CPP = cl
CXX = cl
CCREATEPCHFLAG =
CPPCREATEPCHFLAG =
CUSEPCHFLAG =
CPPUSEPCHFLAG =
FIRSTC = WAVE.C
FIRSTCPP =
RC = rc
CFLAGS_D_WDLL = /nologo /G2 /W3 /Zi /ALw /Od /D "_DEBUG" /FR /GD/Fd"WAVE.PDB"
CFLAGS_R_WDLL = /nologo /W3 /ALw /Ox /D "NDEBUG" /FR /GD
LFLAGS_D_WDLL = /NOLOGO /NOD /NOE /PACKC:61440 /ALIGN:16/ONERROR:NOEXE /CO /MAP:FULL
LFLAGS_R_WDLL = /NOLOGO /NOD /NOE /PACKC:61440 /ALIGN:16/ONERROR:NOEXE /MAP:FULL
LIBS_D_WDLL = oldnames libw ldllcew commdlg.lib olecli.lib olesvr.lib shell.lib
LIBS_R_WDLL = oldnames libw ldllcew commdlg.lib mmsystem.lib
RCFLAGS = /nologo
RESFLAGS = /nologo
RUNFLAGS =
DEFFILE = WAVE.DEF
OBJS_EXT =
LIBS_EXT =
!if "$(DEBUG)" == "1"
CFLAGS = $(CFLAGS_D_WDLL)
LFLAGS = $(LFLAGS_D_WDLL)
LIBS = $(LIBS_D_WDLL)
MAPFILE = nul
RCDEFINES = $(D_RCDEFINES)
!else
CFLAGS = $(CFLAGS_R_WDLL)
LFLAGS = $(LFLAGS_R_WDLL)
LIBS = $(LIBS_R_WDLL)
MAPFILE = nul
RCDEFINES = $(R_RCDEFINES)
!endif
!if [if exist MSVC.BND del MSVC.BND]
!endif
SBRS = WAVE.SBR


WAVE_DEP = c:\source2\cool\wave\filters.h


all:    $(PROJ).DLL $(PROJ).BSC

WAVE.OBJ:     WAVE.C $(WAVE_DEP)
```

```
        $(CC) $(CFLAGS) $(CCREATEPCHFLAG) /c WAVE.C

WAVE.RES:    WAVE.RC $(WAVE_RCDEP)
        $(RC) $(RCFLAGS) $(RCDEFINES) -r WAVE.RC


$(PROJ).DLL::WAVE.RES

$(PROJ).DLL::WAVE.OBJ $(OBJS_EXT) $(DEFFILE)
        echo >NUL @<<$(PROJ).CRF
WAVE.OBJ +
$(OBJS_EXT)
$(PROJ).DLL
$(MAPFILE)
c:\msvc\lib\+
c:\msvc\mfc\lib\+
c:\msvc\lib\+
$(LIBS)
$(DEFFILE);
<<
        link $(LFLAGS) @$(PROJ).CRF
        $(RC) $(RESFLAGS) WAVE.RES $@
        @copy $(PROJ).CRF MSVC.BND
        implib /nowep $(PROJ).LIB $(PROJ).DLL

$(PROJ).DLL::WAVE.RES
        if not exist MSVC.BND    $(RC) $(RESFLAGS) WAVE.RES $@

run: $(PROJ).DLL
        $(PROJ) $(RUNFLAGS)


$(PROJ).BSC: $(SBRS)
        bscmake @<<
/o$@ $(SBRS)
<<
```

Myfilter.c C Source File

```c
#include <windows.h>   // standard Windows header file
#include <mmsystem.h>  // the multimedia header file (youmay not need this)
#include "filters.h"   // header for any filter you build
#include "resource.h"  // resource IDs used for a dialog box if one used

// These two structures are for you to put all "global" variables you will
// be using during the load or save processes.  All vars must go here to
// allow multiple users to use this DLL at once without getting their
// data mixed up.

typedef struct output_tag  // any special vars associated with output file
{      HANDLE hFileOut;  // file handle to output stream
       DWORD dwOptions;    // various options from dialog box ifused
} MYOUTPUT;

typedef struct input_tag // any special vars associated with input file
{      HANDLE hFileIn;   // filehandle to input stream
       DWORD dwOptions
} MYINPUT;

/***************************************************************************/
//     LibMain
//
//     PURPOSE:  Standard LibMain called from LIBENTRY.OBJ
//
/***************************************************************************/

int FAR PASCAL LibMain(HANDLE hModule, WORD wDataSeg, WORDcbHeapSize, LPSTR
lpszCmdLine)
{
    return 1;
}


/***************************************************************************/
//     WEP
//
//     PURPOSE:  No tasks to unload.  Returns success.
//
/***************************************************************************/

int FAR PASCAL __export _WEP (int bSystemExit)
{
    return(1);
}


// If you create a dialog box, here is a sample message proc.
// the important thing is that the nDialogReturn variable gets set with
// your new options if "OK" was hit.

static long nDialogReturn=0L;    // the only globalvariable, used hold the
                                 // options chosen by the user

// This dialog box simply returns a 1, 2, or 3 in nDialogReturn depending on
// which radio button the user chose.

BOOL FAR PASCAL __export DIALOGMsgProc(HWND hWndDlg, WORDMessage, WORD wParam, LONG
lParam)
{switch(Message)
```

```
    {case WM_INITDIALOG:
        if (nDialogReturn==3)
        {   CheckDlgButton(hWndDlg,IDC_RADIO3,TRUE);
        }
        else if (nDialogReturn==2)
        {   CheckDlgButton(hWndDlg,IDC_RADIO2,TRUE);
        }
        else
            CheckDlgButton(hWndDlg,IDC_RADIO1,TRUE);
        break; /* End of WM_INITDIALOG                          */

    case WM_CLOSE:
        /* Closing the Dialog behaves the same as Cancel        */
        PostMessage(hWndDlg, WM_COMMAND, IDCANCEL, 0L);
        break; /* End of WM_CLOSE                                */

    case WM_COMMAND:
        switch(wParam)
          {case IDOK:
              {
               if (IsDlgButtonChecked(hWndDlg,IDC_RADIO1))
               {   nDialogReturn=1;
               }
               else if (IsDlgButtonChecked(hWndDlg,IDC_RADIO2))
               {   nDialogReturn=2;
               }
               else if (IsDlgButtonChecked(hWndDlg,IDC_RADIO3))
               {   nDialogReturn=3;
               }
               EndDialog(hWndDlg, TRUE);
              }
                break;
            case IDCANCEL:
                /* Ignore data values entered into the controls    */
                /* and dismiss the dialog window returning FALSE    */
                EndDialog(hWndDlg, FALSE);
                break;
          }
        break;    /* End of WM_COMMAND                             */
    default:
        return FALSE;
    }
 return TRUE;
} /* End of DIALOGSMsgProc                                        */



// Fill COOLQUERY structure with information regarding this file filter

int FAR PASCAL __export QueryCoolFilter(COOLQUERY far * cq)
{   lstrcpy(cq->szName,"Windows Waveform");
    lstrcpy(cq->szCopyright,"written by David Johnston");
    lstrcpy(cq->szExt,"WAV");
    cq->lChunkSize=30000;
    cq->dwFlags=QF_RATEADJUSTABLE|QF_CANSAVE|QF_CANLOAD|QF_READSPECIALLAST|
QF_WRITESPECIALLAST|
                        QF_HASOPTIONSBOX;
    cq->Stereo8=0xFF; // supports all rates of stereo 8
    cq->Stereo16=0xFF;
    cq->Stereo24=0xFF;
    cq->Stereo32=0xFF;
    cq->Mono8=0xFF; // supports all rates of stereo 8
    cq->Mono16=0xFF;
```

```
                cq->Mono24=0xFF;
                cq->Mono32=0xFF;
                cq->Quad32=0xFF;
                cq->Quad16=0xFF;
                cq->Quad8=0xFF;

                return C_VALIDLIBRARY;
}

long FAR PASCAL __export FilterGetFileSize(HANDLE hInput)
{       long lSize=0L;
        if (hInput)
        {       MYINPUT far *mi;
                mi=(MYINPUT far *)GlobalLock(hInput);
                lSize=mi->lSize;
                GlobalUnlock(hInput);
        }
    return lSize;
}

BOOL FAR PASCAL __export FilterUnderstandsFormat(LPSTR filename)
{       int nFile;
        BOOL bUnderstands=FALSE;

        // The example shown here is for WAV files, but your files
        // will have a different extension, and different format.
        // So you need to check if the header is at the start
        // of the file that you are writing this filter for

        // First check to see if the file extension is "WAV"
        if ((filename[lstrlen(filename)-4]=='.') &&
            (filename[lstrlen(filename)-3]=='W') &&
            (filename[lstrlen(filename)-2]=='A') &&
            (filename[lstrlen(filename)-1]=='V')     )
        {       return TRUE;
        }

        // If not, maybe it had the wrong extension, open the file
        // and see if it has a proper WAV header
        nFile=_lopen(filename,READ);
        if (nFile!=-1)
        {   char m[13];
                if (_lread(nFile,m,12)==12)
                {   if ((m[0]=='R') && (m[1]=='I') && (m[2]=='F') && (m[3]=='F'))
                                if ((m[8]=='W') && (m[9]=='A') && (m[10]=='V')
&&(m[11]=='E'))
                                {       bUnderstands=TRUE;
                                }
                }
                _lclose(nFile);
        }
        return bUnderstands;
}

// This is pretty much the way your procedure should look
DWORD FAR PASCAL __export FilterGetOptions(HWND hWnd, HINSTANCEhInst, long lSamprate,
DWORD dwOptions) // return 0 if no options box
{       int nRc;
        FARPROC lpfnDIALOGMsgProc;
        // our dialog message box proc is defined as ordinal "@20"in the DEF file
        lpfnDIALOGMsgProc = GetProcAddress(hInst,(LPCSTR)MAKELONG(20,0));
        if (dwOptions==0)
                nDialogReturn=1;
```

```
        else
                nDialogReturn=dwOptions;

        nRc =DialogBox((HINSTANCE)hInst,(LPCSTR)MAKEINTRESOURCE(IDD_COMPRESSION),
(HWND)hWnd, (DLGPROC)lpfnDIALOGMsgProc);

        if (!nRc)
            return 0;
        else
                return nDialogReturn;
}

// This is pretty much the way your procedure should look
DWORD FAR PASCAL __export FilterOptions(HANDLE hInput)
{       MYINPUT far *mi;
        DWORD options;
        if (!hInput) return 0;
        mi=(MYINPUT far *)GlobalLock(hInput);
        options=mi->dwOptions;
        GlobalUnlock(hInput);
        return options;
}

DWORD FAR PASCAL __export FilterCanWriteSpecial(HANDLE hOutput,WORDidType)
{       // You provide the code here (see description for this function)
}

DWORD FAR PASCAL __export FilterCanReadSpecial(HANDLE hInput, WORDidType)
{       // You provide the code here (see description for this function)
}

DWORD FAR PASCAL __export FilterWriteSpecial(HANDLE hOutput, WORDidType, unsigned char
far *buf, DWORD dwSize)
{       // You provide the code here (see description for this function)
}

DWORD FAR PASCAL __export FilterReadSpecial(HANDLE hInput, WORDidType, unsigned char
far *buf, DWORD dwSize)
{       // You provide the code here (see description for this function)
}

HANDLE FAR PASCAL __export OpenFilterOutput(LPSTR lpstrFilename,longlSamprate,WORD
wBitsPerSample,WORD wChannels,long lSize, long far *lChunkSize,DWORD options)
{       // You provide the code here (see description for this function)
}

DWORD FAR PASCAL __export WriteFilterOutput(HANDLE hOutput, unsignedchar far *buf,
long lBytes)
{       // You provide the code here (see description for this function)
}

void FAR PASCAL __export CloseFilterOutput(HANDLE hOutput)
{       // You provide the code here (see description for this function)
}

// return handle that will be passed in to close, and write routines
HANDLE FAR PASCAL __export OpenFilterInput(LPSTR lpstrFilename,long far*lSR,WORD far
*wBPS,WORD far *wC,HWND hWnd, long far *lChunkSize)
{       // You provide the code here (see description for this function)
}

DWORD FAR PASCAL __export ReadFilterInput(HANDLE hInput, unsigned charfar *buf, long
lBytes)
```

```
{       // You provide the code here (see description for this function)
}

void FAR PASCAL __export CloseFilterInput(HANDLE hInput)
{       // You provide the code here (see description for this function)
}
```

File Filter API

int FAR PASCAL __export QueryCoolFilter()

COOLQUERY far * cq   This structure must be filled in with all information pertaining to the file filter.

Below is a description of the fileds of COOLQUERY:

char szName[24]	Textual description of file filter that will show in the File Open and File Save dialog boxes.
char szCopyright[80]	Any copyright information you care to put in for your DLL
WORD Quad32	Use the R_xxxx constants ORed together to form the sample rates supported by this filter. Quad (4-channel, 3-D encoded waveforms) are not very popular yet. The '32' means 32-bit samples
WORD Quad16	4-Channel, 16-bit samples
WORD Quad8	4-Channel, 8-bit samples
WORD Stereo8	Stereo, 8-bit samples
WORD Stereo12	Stereo, 12-bit samples
WORD Stereo16	Stereo, 16-bit samples
WORD Stereo24	Stereo, 24-bit samples
WORD Stereo32	Stereo, 32-bit samples
WORD Mono8	Mono, 8-bit samples
WORD Mono12	Mono, 12-bit samples
WORD Mono16	Mono, 16-bit samples
WORD Mono24	Mono, 24-bit samples
WORD Mono32	Mono, 32-bit samples
DWORD dwFlags	Use the QF_xxxx flags ORed together to provide special information about what functions are supported by your filter, etc. and so on. See below for a list of QF_xxxx constants.
char szExt[4];	3-character extension, in caps, followed by a NULL character
long lChunkSize;	Size of chuncks prefered (not really used)

The following constants may be ORed together to form the supported rates for each of the mono, stereo, and quad mode words:

R_5500	Low quality, 5500 Hz
R_11025	AM radio quality, 11 KHz
R_22050	Radio quality, 22 KHz
R_32075	FM radio quality, 32 KHz (also valid for 32000 Hz)
R_44100	CD quality, 44.1 KHz
R_48000	DAT quality, 48 KHz
R_88200	Ultra high quality
R_96000	Ultra sonic high quality
R_132300	Super Duper high quality
R_176400	Extra super duper super high quality

The following are constants that may be ORed together to form the flags dwFlags field:

QF_RATEADJUSTABLE	File filter can handle adjustable rates, not just the standards
QF_CANSAVE	File filter supports saving (writing)
QF_CANLOAD	File loading (reading) supported
QF_UNDERSTANDSALL	File filter can read any file at all (used for PCM)
QF_READSPECIALFIRST	Special information should be read before data
QF_READSPECIALLAST	Special information should be read after data
QF_WRITESPECIALFIRST	Special information should be written before data
QF_WRITESPECIALLAST	Special information should be written after data
QF_HASOPTIONSBOX	This format supports multiple options

long FAR PASCAL __export FilterGetFileSize()
HANDLE hInput   Handle to file information as returned from OpenFilterInput()

You should return the number of bytes that can be read from this file for audio data.   This number can be figured out during the OpenFilterInput() procedure and saved in the input data structure, whose handle you are given.

BOOL FAR PASCAL __export FilterUnderstandsFormat()
LPSTR filename   File name of file to test for understanding of.

You should return TRUE if your file filter can read this file, or FALSE otherwise.   You may need to open the file and read in the first few bytes to see if they are correct for your format.   Or, you may just check the filename extension to determine file validity.

DWORD FAR PASCAL __export FilterGetOptions()
HWND hWnd   Parent window for dialog box

HINSTANCE hInst	Instance handle to use when creating dialog box
long lSamprate	Sample rate of file wanting to be saved
WORD wChannels	Channels in file wanting to be saved
WORD wBitsPerSample	Bits per sample (8,16,...) of file wanting to be saved
DWORD dwOptions	Default options to use, or zero for you to use your own defaults.

You should return zero if no options box is supported.   Otherwise, you should call up an options dialog box with DialogBox() and return a DWORD containing all the information you will need to determine the options chosen.   The DWORD will be passed into the OpenFilterInput() procedure at file read time.

DWORD FAR PASCAL __export FilterOptions()
HANDLE hInput   Handle to file information as returned by OpenFilterInput()

You should return the options settings for the file that has been opened for input.   This is so that a "Save Immediate" can save changes to a file in the same format as the file was originally, without the need to bring up the options dialog box in FilterGetOptons().

DWORD FAR PASCAL __export FilterCanWriteSpecial()
HANDLE hOutput Handle to file information as returned by OpenFilterOutput()
WORD idType   One of the special file types (see below)

Return zero if cannot write the special type, or 1 if the filter can write the type.   None of these special types needs to be implemented, in which case, QF_CANREADSPECIAL, and the other flags should not be set in the dwFlags field of the query structure.

DWORD FAR PASCAL __export FilterCanReadSpecial()
HANDLE hInput   Handle to file information as returned by OpenFilterInput()
WORD idType   One of the special file types (see below)

Return zero if filter cannot read the special type, or the number of bytes that should be asked for if it can read the type.   For SP_LTXT, SP_PLST and SP_CUE types, the number of items should be returned instead.   For SP_NOTE and SP_LABL, the LOWORD of the return value should be the size in bytes of all the zero terminated strings combined, and the HIWORD of the return value should be the number of zero terminated strings in all.

DWORD FAR PASCAL __export FilterWriteSpecial()
HANDLE hOutput Handle to file information as returned by OpenFilterOutput()
WORD idType   One of the special file types (see below)
unsigned char far *buf   Data to be written
DWORD dwSize   Size of data to be written

Return the number of bytes actually written, or (DWORD)-1 if there was an error.

DWORD FAR PASCAL __export FilterReadSpecial()
HANDLE hInput   Handle to file information as returned by OpenFilterInput()
WORD idType   One of the special file types (see below)
unsigned char far *buf   Buffer for data to be read into
DWORD dwSize   Maximum number of bytes to read, maximum.

Return the number of bytes actually read, or (DWORD)-1 if there was an error.

Formats of special data
The following types are single zero-terminated strings.   The length of the string, including the zero terminator, should be returned from the FilterCanReadSpecial() and FilterReadSpecial() operations.   The number of bytes, including the zero terminator, will be passed in as the number of bytes to write in the FilterWriteSpecial() operation.

```
#define SP_IART  20
#define SP_ICMT  21
#define SP_ICOP  22
#define SP_ICRD  23
#define SP_IENG  24
#define SP_IGNR  25
#define SP_IKEY  26
#define SP_IMED  27
#define SP_INAM  28
#define SP_ISFT  29
#define SP_ISRC  30
#define SP_ITCH  31
#define SP_ISBJ  32
```

#define SP_ISRF   33

The SP_DISP type contains data formatted exactly as it would be for the clipboard (CF_xxxx) formats in Windows.   Currently, COOL supports the CF_BITMAP and CF_TEXT formats.   The total number of bytes for the bitmap or text should be returned, and will be passed in.

#define SP_DISP   34

typedef DWORD   FOURCC

struct cue_type
{   DWORD dwName;
         DWORD dwPosition;
         FOURCC fccChunk;
         DWORD dwChunkStart;
         DWORD dwBlockStart;
         DWORD dwSampleOffset;
}

#define SP_CUE   40   // return number of cues of size cue_type
#define SP_LTXT   41   // return number of adtl texts of size 20 (4,id and 4,len, 12 other=0)
#define SP_NOTE   42   // return LOWORD=size, HIWORD=number of strings (sz sz sz...)
#define SP_LABL   43   // return LOWORD=size, HIWORD=number of strings (sz sz sz...)

struct play_type
{   DWORD dwName;
         DWORD dwLength;
         DWORD dwLoops;
}

#define SP_PLST   44   // return number of playlist entries size play_type

HANDLE FAR PASCAL __export OpenFilterOutput()
LPSTR lpstrFilename   File name of file to open for writing
long lSamprate   Sample rate in samples/second
WORD wBitsPerSample   Bits in one sample (8, 16, ...)
WORD wChannels   Channels (1 or 2)
long lSize   Entire size in bytes of audio data that will be written
long far * lChunkSize   Must return the size of blocks you want passed into WriteFilterOutput()
DWORD options   If options used, this is the DWORD containing the option choices

You should open the file for writing, and set up any internal data structures you will be using.   You should return a handle to this data structure to be used by your other procedures in identifying the file and context in which the data is being saved.   In other words, instead of using global variables, put all your variables into one structure, and return a handle to that structure.

DWORD FAR PASCAL __export WriteFilterOutput()
HANDLE hOutput   Handle returned by OpenFilterOutput()
unsigned char far *buf   Buffer containing data to be written
long lBytes   Number of bytes to write

The number of bytes to write will most generally be the size of the chunk given back through OpenFilterOutput().   Return the number of bytes actually written.

void FAR PASCAL __export CloseFilterOutput()
HANDLE hOutput Handle returned by OpenFilterOutput()

You should clean up, and close the file.   As far as COOL is concerned, the file has been completely written.

HANDLE FAR PASCAL __export OpenFilterInput()
LPSTR lpstrFilename   File name of file to read
long far *lSR   Sample rate, to be filled in on file open.
WORD far *wBPS   Bits per sample, should also be filled in by you.
WORD far *wC   Number of channels (1=mono, 2=stereo) to be filled in by you.
HWND hWnd   Handle to a Window (for printing error MessageBoxes()?)
long far *lChunkSize   Must return the size of blocks you would like to have passed to ReadFilterInput().

DWORD FAR PASCAL __export ReadFilterInput()
HANDLE hInput   Handle returned by OpenFilterInput()
unsigned char far *buf   Buffer to place data into

long lBytes   Number of bytes (maximum) to read into buffer

Return the actual number of bytes placed into the buffer.

void FAR PASCAL __export CloseFilterInput()
HANDLE hInput Handle returned by OpenFilterInput()

Done reading file, please close and clean up.