A*wave* *v.*4.1

***A**udio &* ***W**avetable* ***I**nstrument*

***C**onverter,* ***E**ditor* *&* ***P**layer !*

Shareware

***W**ritten by:*

F. Markus Jönsson
mail: Svedjevägen 7, 147 52 Tumba, Sweden
www site: http://hem.passagen.se/fmj/fmjsoft.html
email: fmj@hem.passagen.se

Terms of Use

FMJ-Software exclude any and all implied warranties, including warranties of merchantability and fitness for a particular purpose.

Nor does *FMJ*-Software make any warranty of representation, either express or implied, with respect to this product, its quality, performance, functionality or fitness for a particular purpose.

Nor shall *FMJ*-Software have any liability for special, incidental, or consequential damages arising out of or resulting from any use of this product.

Nor shall *FMJ*-Software have any responsibilities, nor any obligations whatsoever to provide support for this product.

Nor shall *FMJ*-Software have any responsibilities whatsoever to provide upgrades or bugfixes of any kind.

✘

Background and Features

Historical background

A*wave* first saw the light of day in late '93 when I needed to extract some samples from an Amiga tracker module and save them as UltraSound instruments. So, I hacked up a small program then called MOD2PAT. I thought that perhaps other people might find it useful so I released it on the Internet. As some programs tend to do, this one soon grew beyond all bounds. In response to user demands, more file formats were continually added (and still are). It was ported to Windows, and it got a steadily improving user interface. The name was now something of an understatement so I changed into 2PAT. It didn't take very long until that name too was somewhat inappropriate so it was changed into *Awave* (*pron. Ae-wae-v*). Today, after several major rewrites, it handles more than one hundred audio carrying file formats for import, as well as most popular formats for export. And sporting a rich set of editing, playback and auditioning functions to boost! Ever since the first days, *Awave* has been developed purely as a hobby project. It would not have been possible without the help and support of several dedicated users. Thank you!

Features

A*wave* reads a veritable host of audio carrying file formats from different platforms, synthesizers and trackers. It can be used in a variety of ways: as an audio file format converter, an audio editor, an audio player, and, last but not least, as a wavetable synthesizer instrument format converter and editor. Data is internally represented in a superset of the format defined by the DownLoadable Sounds architecture level 1 standard. It can be viewed as a graphical tree with the 'collection' as the stem, instruments, layers, and regions as branches, and finally, waveforms as leaves.
If you have one of the supported wavetable synthesis sound cards you can simply click on an item, and directly 'audition' it using either a midi keyboard or the computer keyboard. If don't have such a card, you are still in luck! There is a crude software synthesizer that runs on any sound card!
Saving to any of the supported export formats is just a button away! Sending and receiving MIDI SDS dumps are also supported for transferring waveforms to professional synthesizers. There are visual editors for all parameters like waveform loop points and layer keymaps. Basic audio processing and editing functions are also provided.
The 'AudioPlayer' feature is a nice audio playback control, sporting both real-time oscilloscope and frequency analyzer graphs. It is available from inside *Awave* as well as integrated into the Windows95 shell in order to easily let you play the sounds of any of the supported file formats! Thus it is also and ideal 'helper app' for your WWW-browser!
If you are into any form of sampled audio on the Windows platform, you'll most likely find this program useful!

Using Awave

Welcome to Awave!

This small text will teach you the basics of using *Awave*. It is highly recommended that you read through it at least once; it may save you both time and frustration later. The text is divided into the following sections:

What can *Awave* do?
Configuration
Converting audio files
Converting instruments
Editing and inspecting

What can Awave do?

A*wave* is in many senses a multi-purpose audio tool.
Here are a few of the cool things that you can use it for:

- Convert from more than one hundred audio carrying file formats into most (dare I say all?) common audio file formats. You can also open, process, and save multiple files if you need to do 'batches' of files.

- Use it as an audio player extension to Windows. It even sports a nice real-time oscilloscope or spectrum analyzer graph!

- Are you a composer? Ever wanted to take advantage of all those free instruments floating around the web but found it impossible to get them in a format suitable for your synthesizer? *Awave* reads the instrument formats of most (if not all) commercial synthesizers and soundcards and can also write a fair number of them! It also supports SDS so you can send samples over Midi.

- Ever wanted to play a midi song that comes with an 'instrument bank' that were created for another wavetable sound card than yours? *Awave* can make *high quality* conversions between the bank and instrument formats of most of the major sound cards!

- Use it as a simple audio editor. You can do such things as resampling, fading in and out, maximizing amplitude, and more.

- Use it as an instrument editor for your wavetable sound card. It may not rival the sound card specific instrument editors since it will not support all the finer peculiar details specific to every card. But it is a very useful editor if you are content with having 'only' the most commonly used (and useful) features at your command. As a bonus you will be able target your creations for most major sound cards! It provides full support for all the features (in fact a superset) of the new DownLoadable Sounds standard.

Configuration

If you have come this far I suppose that you have already read the README.TXT file and run the SETUP.EXE program, right? Anyway, if ever you have trouble with something in *Awave*, be sure to browse through the FAQ section of this help file for an answer. And if ever your wonder about what exactly a certain control does, try selecting it and then press F1.

Very well, let's do some configuring!

Start by selecting Program setup... from the Options menu to enter the configuration dialog.

The first thing you see is the General tab. Here are several things but nothing much that we are interested in right now. Remember that you can always use the [?] button and click on anything that looks interesting to get more information about it.

Select the AudioPlayer tab. The 'AudioPlayer' refers to *Awave's* audio playback engine. It is usually seen in the form of a special audio playback window but is also used by the *Awave* software synthesizer (more on that later).
Under Primary dev:, select your 'primary' audio playback device**.** When playing audio, *Awave* will always first try to use this device, if it is 'busy' playing something else at the moment, then any secondary device(s) that you may have installed will be used instead (i.e. if you are a happy owner of multiple sound cards).
Next, let's select the 'method' that should be used for audio playback. If you have *Microsoft DirectX* (v2 or later) installed on your computer then it is highly recommended that you select the '*Use Microsoft DirectSound services*' option. This will give much smaller delays when playing sounds and will also make the *Awave's* software synthesizer sound much, much better... If you do not have *DirectX* installed (or perhaps you are using Windows 3.1, for which DirectX is not available) then select '*Use Windows multimedia services, buffered output*'. If (and only if) you have problems with no sound in the AudioPlayer try '*Use Windows multimedia services, unbuffered output*'.

Now select the Auditioning tab. Audition refers to a feature that will let you directly play an instrument or an audio sample in *Awave* using an external midi keyboard or even your computer keyboard. To enable it, simply check the Enable auditioning box.
In the MIDI-Input list, choose what device you want to use for 'playing' on. If you don't have, or don't want to use, a proper midi-in device, then select <none> from this list. If you check the Enable virtual keyboard box, you can use the computer keyboard as a midi keyboard.
In the Wave-synth list, select any of the wavetable synthesizers that *Awave* provides 'native' support for (i.e. if you happen to have one of them) or else, simply select the *Awave software synthesizer*.
The drawback(s) with having auditioning enabled is that it does slow down the program somewhat when selecting items. Also, while it is on, you may not be able to simultaneously use your 'Wave-synth' in other applications. Therefore, the program toolbar contains a handy 𝔍 button for quickly toggling this feature on and off.
More information about auditioning support is found here.

Finally, select the Helper apps tab. It is possible to use an external (specialized) application of your choice for recording, editing, or otherwise processing waveforms. In order to make that possible you must enter the program name and path for the editor under the Waveform editor heading. You can easily browse the disk for it by pressing the [..] button. Then enter any command line switches that you want the program to start with, e.g. "c:\foo\myeditor.exe –load". A .WAV file name will be appended to the end of this string when starting the editor so if it can't open a .WAV file from 'the command line' then it can't be used with *Awave*… As default, the simple 'Sound Recorder' that comes with Windows is selected but you may want to change it to a more capable program.
There are also fields for selecting a sound mixer/ volume control app and an on-screen keyboard app

(e.g. for use with the auditioning feature). These two can then be launched by a simple key presses or by selecting an entry from the main Options menu.

Ok, we're done configuring, so click OK!

Converting audio files

The basic building block in *Awave* is the waveform. You may also have seen it called it a 'sample', a 'sound', 'audio', or something similar in other programs and in literature. In strict *Awave* terminology :-), a waveform is a temporal sequence a wavesamples. A wavesample in turn is a number describing a finite 'sample' of a sound (it can be either mono or stereo). The waveform is the actual recorded and digitized sound that is played back by your computer. It is also the basic building block of so called wavetable synthesizers. A basic 'audio file' is a computer file that contains one single waveform plus perhaps some auxiliary data describing the sample rate and such things.

To open (import) an audio file, press 📂 on the toolbar or select *File->Open new* from the program menu and select a file using the dialog box. As you can see in the bottommost status bar, *Awave* will try to recognize the correct <u>import format</u> of a file even though it may have an incorrect file extension. If you want, you may select several files to open them all at once. This comes in very handy for doing 'batch' conversions. To later open more files without throwing away the presently loaded data, use the *File->Merge with another file* command. The number of files you can simultaneously have open in this way is limited by the amount of memory in your computer. Since *Awave* will never load any actual waveform data until it is absolutely necessary (lazy loading), you may want to keep this in mind and refrain from doing 'mega big batches'. Else it may just drop out with an out of memory message in the middle of the batch. Do somewhat smaller batches instead if you must.
Noticed the Formats… button? Pressing it will bring you *Awave*'s help reference containing short descriptions about all supported file formats.

Anyway, click OK in the open file dialog to load the file(s) that you have selected. You should now see an expanded
📁 folder containing
✛ or
✛ items for the waveforms that you just loaded. With an item I shall in the future mean any 'object' in this 'tree view list'. The topmost
� item is called the collection and is sort of a container for all object that you have currently loaded.

To play one of the waveforms you have just loaded, select it and press
▶ or use the corresponding Play audio menu command. This will bring up the AudioPlayer window that I talked about in the configuration section. To close the AudioPlayer click the >>Next>> button or the [X] button. The other controls should be self-explanatory. Hint! If you want to play all waveforms in a row, select and play an item 'one step higher up' in the tree-view list hierarchy. Then use >>Next>> to proceed to proceed to the next waveform. To do more advanced editing press
📋 or select Properties from a menu. More on that in the <u>editing</u> section.

Now let's try saving (exporting) and audio file!
Select the waveform that you want to save and then press
💾 or select File->Save waveform as from a menu. Notice that there is no 'regular' save command in *Awave*. You must always first select the particular item that you want to save and then issue a save command for it.
Anyway, this brings up a save file dialog where you can type in file name et c. as usual. In the File format

list you select what file format you want to save it in. There is also a Subformat list which changes depending on what file format you have chosen. There you can sometimes select what sub format of the selected file format that it should be saved in. Typical choices are 8bit or 16bit data formats. Waveforms are always stored in 16bit format internally so even if the original data were 8bit, 16bit will be selected as default. You can of course manually override this by selecting 8bit in the subformat list. Next, mark the Mono and Stereo checkboxes to indicate what channel format you want to save it in. If both boxes are checked (only possible with file formats supporting both stereo and mono of course) then it will be saved in the channel format of the original waveform i.e. 'leave it as it is'. This is the default selection and probably is what you usually want to do.

Now press OK to save the waveform or Cancel to skip it.

If you want to do a 'batch save' then you should use the File->Save all in multiple files command instead. This first brings up a dialog where you can choose how the output files should be named. It is not self-evident how this should be done so you are presented with a choice of a few 'naming schemes'. Next press Continue. A new dialog comes up, looking just like the normal 'single save' dialog. Here you can enter a 'base name' for the naming scheme that you selected. You can also select the file format and subformat that you want to save to. All of the output files will share these selections. *Note!* If you need to use old 8.3 file names you must yourself ensure that the naming 'scheme' that you have selected will not give too long names.

Converting instruments

To read a file containing an instrument or a collection of instruments (often called a bank file), follow the same procedures as described for reading audio files. The same things also apply regarding opening and saving multiple files in 'batches'. You can also read so-called 'tracker files. These are files containing both a collection and a 'song' that uses the instruments in the collection. For those files *Awave* will *only* read the instruments and waveforms. The song is *not* converted.

A tree-view list with a ✖ Collection to

▤ Instrument to

♠ Layer to

♠ Region to

✖ Waveform, hierarchy will be constructed from the file. These data items is a superset of the DownLoadable Sounds (DLS) Level 1 standard. Please see the appropriate DLS documentation for a detailed description. Some other symbols may also appear:

✖ - Folder containing all waveforms,

✖ - Stereo waveform. If the file in question do not contain any instrument data (as in the case of a pure audio file), a collection will be created with a folder containing the one waveform (representing the audio data).

Select an instrument or a waveform from the list by clicking on it with the mouse.
You may now be able to play it on your midi keyboard (or computer keyboard if you have enabled the virtual keyboard feature), provided that you have setup auditioning properly. See <u>configuration</u> and <u>auditioning support</u> for details. Or, you can press
▶ or use the *Play audio...* menu item to play the sound using the AudioPlayer.

To save an item, first select it (
✖,
▤ or
✖), then use

✖ on the toolbar or press the right mouse button and select *Save X as...*, or select the corresponding command from the File menu. *X* will reflect the type of currently selected item (if nothing is selected it will default to 'collection'). A standard dialog box will appear and prompt you for the file name to save to. Same thing goes as for saving waveforms. The selections made will apply to *all* data stored in the output file.

Note! Many of the export formats only support saving waveforms (and not collections et c.), see the symbols in the export formats list. Moreover, the capabilities of the various 'instrument capable' file formats and synthesizers may vary greatly, see notes on cross platform compatibility on how to create instruments that can be used on as many wavetable sound cards as possible. An especially thorny issue is how to best convert articulation data (lfos, envelopes…) between formats for different synths with different capabilities. There are some notes in the FAQ that explain some important issues relating to this that may want to read once you have mastered the basics of this program.

Editing and inspecting:

To edit the name of an item, do a 'slow double click' on the name or press F2 and edit the name as in a standard text edit control (same as when editing filenames in the Windows 95 shell).

For all other parameters, select the item and then press ✖ on the toolbar or select Properties from the right mouse button menu, in order to bring up its property sheet. The right mouse button menu will also contain a list other useful commands. Try this for all the various item types (collection, instruments, layers, regions, waveforms, …) and carefully explore all the tabs! You will find, among other things, a visual loop point editor, a waveform editor, a layer keymap editor and an audio processor wizard! As mentioned earlier, *Awave* supports a superset of the DownLoadable Sounds level 1 architecture and all associated parameters. I must refer to the DLS documentation for a detailed description of all parameters. When this was written this documentation has not been formally released yet but this will happen in the near future.

Note! When auditioning is enabled, you can click the Apply button in a property sheet to download the changes then hear them directly by playing them on your midi in device.

If you turn on the Options->Auto property sheet, then there will always be a property sheet open, showing the properties for the *currently selected* item.

When doing 'batch conversions' you may want to do a lot of audio processing and stuff on all items and would not want to manually go through the hoops of the property sheets for every item. For this purpose there is the *Edit-> Process all...* command. It will bring up a dialog that will let you apply certain very common operations to all items. Things that you can do here include:

- Convert all stereo waveforms into mono.
- Split all stereo waveforms into dual mono waveforms and convert any layers using them into 'dual layers'; one layer panned full left using the left mono waveforms and another corresponding layer for the right. Practically no wavetable synthesizers can handle real stereo but uses 'dual layered' mono waveforms to achieve the same affect. Thus this option will be useful if you want to use stereo waveforms in your instruments. If you do *not* do this manually, then when you save an instrument containing a 'real' stereo waveform to a file format that does not support it, it will automatically be converted to and saved as mono (with the two channels mixed into one).
- Compact multi-layered instruments by merging any non-colliding layers. Some synthesizers have a definite limit on how many layers they can use (e.g. four layers in TB WaveFront cards) so this feature may come in handy for instruments that use a lot of unnecessary layering…
- Compact then split-multi-layer instruments into several single layered instruments. This is useful if you want to save in an instrument file format for a synth that do not support layers (e.g. UltraSound

GF1 cards). If you do not do this manually, only the first layer in multi-layered instruments will be saved.

- Resample all waveforms to a common sample rate.
- Convert the loop format. Some synthesizers do not support reverse or bi-directional loops. If you try to save to a file format that do not support a certain loop type, it will be saved as forward loops with no further conversion. This may sound awful. Using this feature you can convert such loops to forward loops by constructing a forward loop that emulates the reverse and/or the bi-directional loop. This will enlarge the waveform(s) a little but will sound correct on all synthesizers. You also have the option to remove everything after the loop end point except some padding bytes to keep 'loop interpolation' happy. This will shrink the waveform(s) by removing this potentially wasted/unused end part.

There are much more 'processing' you can do on individual wave using *Awave's* 'audio processing wizard'. To access it, select the waveform, then select properties and go to the Editor tab. Press the Process button. to get a list of available options. Many of the options in the list only turn up if you first make a 'selection' in the editor page. To do a selection, you hold down the left mouse button drag with the mouse over the wavesample range that you want. Alternatively, you can press the Select all button to make a selection covering the whole waveform. Other options may appear or not depending on various characteristics of the waveform, e.g. 'convert to stereo' only appears for mono waveforms. An example of a useful audio processing option is 'Crossfade loop' that will let you crossfade' the wavesamples at the end of the loop with the wavesamples just prior to the loop. This is an easy way to create good sounding loops without much work. Hint! Too get good sounding loops follow this recipe: Go to the loop editor. Try to create a decent looking loop (it doesn't have to sound perfect, e.g. the amplitude of the 'tops' at one side of the loop split may be much different than on the other side). Be *very careful* that there is an 'integer' number of 'periods' in the loop, i.e. that the distance between 'zero crossing's is about the same, otherwise you may get very bad 'interference' effects. Then switch to the editor tab but *first click 'Apply'* in the loop tab, or you will may loose the newly set loop points. Then click process->crossfade->finish. Enjoy your new loop!

If you need even more waveform editing and processing power, or if you want to record new waveforms, then you can use an external, specialized, helper application for this purpose. There are numerous such programs on the market, shareware as well as commercial. You can select which app to use in the program setup dialog. To use it, select a waveform and select Run editor app from the right mouse button menu or Waveform editor from the Edit menu. The selected waveform will then be saved to a temporary file, "Awave.wav", on the disk and the external editor will be launched with this file as a command line parameter. While you use the external editor, *Awave* will 'freeze up'. When you are done editing, save it and close the external editor. *Awave* will then 'unfreeze ', reload the waveform, and delete the temporary file. To record a new waveform, select File->Add new waveform, then run the editor/recording app on this waveform and record a new one 'over' the old.

*Now you should have a general idea of how **Awave** works... There are still many features that have not been described in the text above. You may wish to continue with the Commands section.*
After that, you may want to glance through the FAQ section of this help file for additional 'hints' on how to use the program! Happy Awaving!

Commands

I apologize for the somewhat sparse documentation regarding menus, property sheets, keyboard shortcuts, command line options et c. Here you will find a brief listing of all such things so that you may see what is available. Most of the commands can be issued in multiple ways, from the main program menu, from the toolbar, from the right mouse button menu, and/or using a keyboard shortcut.

The menus

File menu:

Open new...	Clear all and open a new file. Key: *F3*
Save X as...	Saves the currently selected item (of type X). Key: *F4*
Merge with another file...	Merges (inserts) the contents of a file with the current bank. Key: *Shift-F3*
	Tip! An alternative way to merge files is to drag them from the Explorer.
Save all in multiple files...	Saves all instruments or waveforms depending on output format. Key: *Shift-F4*
Autocreate instruments	Creates instruments for all 'lonely' waveforms.
Add new instrument	Create a new, empty, instrument in the current bank.
Add new waveform	Create a new, empty, waveform in the current bank.
Exit	Close the program. Key: *Alt-X*

Edit menu:

Properties...	Will pop up a property sheet information and editors. Key: *F5*
Play audio	Launches the integrated *Awave* audio player. Key: *F6*
Process all...	Process / convert all items in one go (Resample, Loops, et c.). Key: *Ctrl-A*.
Copy -> clipboard	Export a waveform to the Windows clipboard. Key: *Ctrl-C*
Paste <- clipboard	Import a waveform from the Windows clipboard. Key: *Ctrl-V*
Midi SDS transmit	Opens the Midi SDS transmit dialog. Key: *Ctrl-T*
Midi SDS receive	Opens the Midi SDS receive dialog. Key: *Ctrl-R*
Edit X name	Edit the name of the selected item (of type X). Key: *F2*
Waveform editor	Run the waveform editor helper with selected waveform. Key: *Ctrl-W*
Delete X	Delete it the selected item (of type X). Key: *Del*
Delete unused waveforms	Deletes all waveforms not used in any instrument.

Options menu:

Program setup...	The program setup property sheets. Key: *Ctrl-O*
Auditioning	A quick switch for toggling auditioning on/off. Key: *F7*
Auto property sheet	Toggle auto property sheet on/off. Key: *F8*

| *Run mixer app* | Run the mixer helper app (if defined). Key: *Ctrl-M* |
| *Run keyboard app* | Run the on-screen keyboard helper app (if defined). Key: *Ctrl-K* |

Help menu:

| *…* | Brings up various help sections. Key: *F1* |
| *About **Awave*** | Brings up some program info. Key: *Ctrl-A* |

Toolbar:

✖ Clears the instrument list and prompts you for a new file.

✖ Saves the currently selected item.

✖ Will pop up a property sheet information and editors.

✖ Launches the integrated *Awave* audio player.

✖ Quick switch for toggling auditioning on/off.

📖 Brings up the content page of this file.

Left mouse button:

Single click	Select an item.
Double-click	Expand or shrink the tree view list.
Slow double-click	Edit name field of item.

Right mouse button menu:
The exact menu depends highly on the selected item type!
Contains the most useful commands for each type of items!

The property sheets

Almost all of the editing power of *Awave* is hidden in property sheet dialogs, easily accessible through the right mouse button or the ✖ toolbar button. The various data items is a superset of the Downloadable Sounds (DLS) Level 1 standard. For detailed description of the various parameters available, please see the appropriate DLS documentation (I will upload this to my www site as soon as it has been released).

Collection properties:

| *General tab*, | Info, comment, copyright notice, |
| *Programs tab*, | Program no assignment to instruments. |

Instrument properties:

General tab,	Type, bank no, program no, comment,
Layers tab,	List of layers,
Articulation tab,	Instrument global articulation enable, panning, lfo,
EG1 tab,	Instrument global volume envelope,
EG2 tab,	Instrument global pitch envelope.

Layer properties:

| *Keymap tab*, | Key to region editor. |
| *Articulation tab*, | Per layer articulation override, panning, lfo, |

EG1 tab,	Per layer volume envelope,
EG2 tab,	Per layer envelope.

Region properties:

General tab,	Waveform link, key & velocity range, key group, loop & root override.
Articulation tab,	Per region articulation override, panning, lfo,
EG1 tab,	Per region volume envelope,
EG2 tab,	Per region envelope.

Waveform properties:

General tab,	Info, sample rate, root-key, volume, pan,
Editor tab,	Waveform editor and display (power or current selectable),
	Audio processing wizard (edit button):
	"Convert bi-directional to forward loop",
	"Convert reverse to forward loop",
	"Convert to stereo channels"
	"Convert to mono channel(s)",
	"Crossfade loop",
	"Cut away selection",
	"Double the loop section length",
	"Fade in/out selection",
	"Insert silence",
	"Maximize amplitude",
	"Remove silent sections",
	"Resample waveform",
	"Reverse selection",
	"Set loop points to selection",
	"Smooth out the selection",
	"Statistics",
	"Trim waveform to selection"
Loop tab,	Visual loop point editor, loop type,
Source tab,	Source data type, file position and size.

✖

AudioPlayer command line interface

If you start *Awave* with -AP as the first option on the command line and follows this with one or more filenames and/or directory names, the Audio Player mode will be entered. You can also launch the Audio Player at any time from the normal main window of the program with the Play audio command..
In this mode *Awave* acts as a digital audio player that can play all the import formats supported by *Awave*.

Files will be consecutively loaded and all the sounds contained within them played before the program terminates. When a directory name is specified, all the recognized audio files in that directory will be played. You must press the >>Next>> button to advance from one waveform to the next, closing the Audio Player after the last one (to immediately close it, skipping any remaining waveforms, click the standard close window button |X|). The play/stop button should be obvious. The *off, wf, sp* radio buttons determines the operational mode for the scope (off/hold, waveform oscilloscope, or frequency spectrum analyzer).

The following command line switches are allowed, and must precede the file names.:
-AP   Specifies that the AudioPlayer mode shall be entered.
-ONCE   Automatically starts playing the next item when the current one is done; closes the AudioPlayer after the last one.
-DELETE   Delete the input file(s) from disk after the AudioPlayer is closed.
-NOWIN   Runs the AudioPlayer and plays the sound without displaying any window.
-LOOP   Repeatedly plays the indicated files and/or directories. Break the loop by pressing the |X| button.
-NOMSG   Turns off all warning and error dialog boxes.
Warning if you combine the -NOWIN switch with the -LOOP switch then *Ctrl-Alt-Delete* is the only way to break the loop.

With the Audio Player option, you can use *Awave* for example as a helper app to your favorite Web-browser!

Example: "AWAVE -AP QUADRA.AU POPCORN.MOD"
Example: "AWAVE -AP -ONCE -NOWIN -DELETE filename"
Example: "start AWAVE -AP -ONCE -LOOP ."
The last example repeatedly plays all files in the current directory (the current directory is, as usual, denoted by a single dot (.)). The 'start' command can be used in Windows 95 to start a 'registered' program without having to have it in the path or having to type whole the path explicitly.

✄

Auditioning support

If you have a:
- Creative Labs SoundBlaster EMU8000 based card (AWE32, SB32, …), or a
- Turtle Beach WaveFront based card (Maui, Tropez, …), or a
- Gravis UltraSound GF1 based card (GUS Classic/MAX/ACE), or a
- Gravis UltraSound InterWave based card (GUS PnP), or
- *Any digital audio card* (using the *Awave* software synth)
you can use the auditioning feature of *Awave* to directly hear and play instruments and waveforms by just selecting them in the main tree-view list.

To enable auditioning, go to the Options->Program->Auditioning tab. Just click the Enable checkbox and choose appropriate MIDI-Input and WaveSynth devices. If you don't have a midi-in device select the <none> midi input device. If you check the Enable virtual keyboard box, you can use the computer keyboard as a midi keyboard as well. Click OK and everything should be go.

The virtual keyboard has the following layout:

```
 _____
||S|D|||G|H|J|||2|3|||5|6|7|||9|0||
|| |_|_|||_|_|_|||_|_|||_|_|_|||_|_||
|Z|X|C|V|B|N|M|Q|W|E|R|T|Y|U|I|O|P|
|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
  |                |               |
 C-1             C 0            C+1

  Shift-Up arrow           Increase note velocity
  Shift-Down arrow         Decrease note velocity
  Ctrl-Up arrow            Increase octave shift
```

```
     Ctrl-Down arrow           Decrease octave shift
```

Note: the `Add` and `Subtract` keys are the – and + keys on the numeric keypad.
If you have a non-standard, or foreign country keyboard layout, then there is a way to <u>edit the key layout</u>.

✖

MIDI SDS transfers:

SDS, or the MIDI Sample Dump Standard defines the standard method for transfer of sound sample data between MIDI equipped devices. A 'sample' here means the same as what *Awave* calls a 'waveform'. Please consult your synthesizers manual to see if it supports MIDI SDS, and if it does, how to put it an an appropriate 'mode' for sending or receiving samples via SDS. Sample dumps may be accomplished with either an **open loop** or **closed loop** system. *Awave* supports both types of systems but the closed loop system is *highly recommended* over the open loop system.
In the **open loop** system, you simply have a cable between the transmitters (senders) MIDI out jack to the receivers MIDI IN jack and do a straight dump of all sample data. There is no 'handshaking' between them at all, which means that there is no error correction and that if the receiver can not keep up with the data (e.g. if it need to flush it to disk) then it will loose data.
In the **closed loop** system, you also have a cable from the receivers MIDI out jack to the transmitters MIDI in jack (i.e. two cables) so that it can receive feedback from the receiver. Thus if a data packet is 'lost' during the transmission, it is automatically resent. It is also faster than the open loop system since in the open loop system the transmitter has to allow 'generous' time for pauses between data packets in order to give every possible receiver a decent chance to process them. In the closed loop system on the other hand, the receiver immediately tells the sender when it is ready for the next packet.

How to send a sample from Awave to a synthesizer using SDS:
1, Connect your MIDI cables properly (connect your computers MIDI out to the synthesizers MIDI in and vice versa).
2, Put your synthesizer in a mode for receiving MIDI SDS (see your synthesizers manual). This is somthimes called 'slave' mode.
3, In *Awave*, select the waveform that you want to transmit. It is also possible to send several waveforms in one 'batch' by instead selecting any one item 'higher up' in *Awave'*s 'heirarchy', e.g. an instrument or the waveforms folder (the latter will let you send all loaded waveforms).
4, Select 'Edit'->'Midi SDS transmit'. The Midi SDS transfer dialog pops up. First select the MIDI in and out devices to which you have connected your synth. Then you may have to select a 'channel'. Some synths uses the channel to determine if they should respond to a certain dump or not, others ignore it. Then enter a 'sample' number. The sample number is used to determine in what 'slot' or 'location' on the receiving synthesizer the sample/waveform will be stored in. The exact interpretation of this differs between different synthesizers.
5, Click 'Send this' to start transmitting the 'current' waveform. The name of the 'current' waveform is always displayed to the right of the sample number. A status bar at the bottom of the dialog displays the progress of the transfer. When it is finished, the name of the next waveform in the 'batch' is displayed. If you only selected one waveform, or when the last waveform in the batch has been sent then the transmit buttons are greyed. To send the next waveform, i.e. the new 'current' waveform, just click 'Send this' again… There is also a button labeled 'Send all' that will simply send all waveforms in a batch without a pausing. Another button labeled 'Skip this' will skip the current waveform in a batch: Using the 'Send this' and the 'Skip this' buttons will let you selectively send only some of the waveforms in a batch.
6, Click the [x] button to close the dialog when you are done.

How to send a sample from a synthesizer to Awave using SDS:

1, Connect your MIDI cables properly (connect your computers MIDI out to the synthesizers MIDI in and vice versa).
2, Put your synthesizer in a mode for transmitting MIDI SDS (see your synthesizers manual). This is somthimes called 'master' mode.
3, Select 'Edit'->'Midi SDS receive'. The Midi SDS transfer dialog pops up. First select the MIDI in and out devices to which you have connected your synth. Now there is two ways to start the transfer:
4, Click the 'Start' button. Then use your synthesizer's control panel to start the transfer (see your synth manual again). *Awave* will responds to transfers on all 'channels' and to all 'sample numbers' so you can leave your synthesizer to its default values. The progress of the transfer will be displayed in the dialogs status bar. When the transfer is completed, it will display that it is ready to receive another sample.
5, The alternate way to start the transfer is to first fill in a sample number in the dialog. Then press the 'Request' button. This will ask the synthesizer to send the sample corresponding to that number and will automatically start the transfer. Not all synthesizers may support such requests though, and it will certainly not work on open loop systems.
6, Click the [x] button to close the dialog when you are done.

Ensuring cross platform compatibility

When authoring instruments and midi songs that should be usable by as wide a range of different wavetable sound cards an synthesizers as possible, follow the following 'guidelines':

- Do not use stereo waveforms.
- Do not use reverse or bi-directional looping.
- Do not expect perfect reproduction of envelopes and other modulation parameters.
- Do not use more than 512K samples and 1MB of waveforms and use no individual waveform > 256KB.
- Remove all unused waveforms from the bank.
- Use only one layer per instrument.
- Us a maximum of 16 regions in an instrument.
- Use only instrument global articulation.
- For regions, always use the complete velocity range; i.e. 0..127.
- Use only one 'bank' and in your songs, do not use any midi program no not defined in that bank.
- In your songs, the drum channel, ch10, is always supposed to be the standard GM drum bank.

If you want to edit the key assignment for the virtual keyboard you can do that with the registry editor (regedit.exe) that comes with *Windows 95*. Open this_computer->HKEY_CURRENT_USER->Software->FMJ-Software->*Awave* and edit the mtVKeys string. It's just a string of (normally 29, but maximum 36) uppercase characters holding the ASCII key to assign to a note, beginning with C3 for the first character and then in increasing order...

Awave can use any audio card capable of playing digital audio to emulate a very simple wavetable synth in software. It should be enough to give a rough idea of an instruments or waveform.

It supports the following midi messages: note on, note off, pitch bend, all notes off.

If you have set 'Use Microsoft DirectSound services' in the AudioPlayer property tab' then the software synth will be 16 voice polyphonic and support key velocity, volume and pan. If not (i.e. in 'Use Windows multimedia services' mode), those parameters are ignored, it will be monophonic, and it will generally sound much worse...

Awave supports downloading instruments to the Advanced Gravis GF1 based synthesizers (UltraSound Classic/Max/Ace soundcards).

Remember that, that a GUS with full memory can handle a maximum of 1M of 8bit samples.

Awave supports downloading instruments to the Advanced Gravis InterWave based synthesizers (the UltraSound PnP soundcard).

You need the sound card drivers v2.0 or later. You also need to have 'IWINIT.EXE' in the same directory as your IW.INI file, or it won't work…

Note! The amount of memory on the soundcard can currently not be detected so the program never knows if an instrument will fit in memory or not. It will always try to load instruments up to 4MB in size (with 16bit samples always used during auditioning) and never any instruments larger than that.

Awave supports downloading instruments to Turtle Beach WaveFront based synthesizers (Maui, Tropez, Rio, ...)

You must have installed the WFGATE.DLL file into your Windows directory in order to use auditioning with the WaveFront synth. This file is normally installed with TB's WavePatch software but you may also find this file on my www site.

Note! Any custom instruments that you may previously have loaded into wavetable RAM will be overwritten. You may also have to reset the synth to restore the GM setup after using *Awave*.

Awave supports downloading instruments to the Creative Labs- EMU 8000 based synthesizers (SB AWE32, SB32, …).

Note! You will need the very latest drivers with 32bit application as well as .SF2 bank file support in order to be able to use auditioning with the AWE32 (these drivers were only available in beta version when this was written). Also make sure that the AWEMAN32.DLL file is correctly installed in your windows directory.

Note! If you run out of memory on the card when trying to load an instrument, try removing any other user banks you may have previously loaded to card memory (by using the standard AWE32 control panel).

Note! User bank 120 will be used as a 'scratch bank' and any data in it will be lost…

Revision History

version 4.1

UltraSound PnP news! Added 'native auditioning' (sample downloading) support as well as support for amplitude envelopes (.FFF files)!

Added 'basic' support for the new Turtle Beach Pinnacle .PBF files. You can read and write 'multi-sampled' as well as 'multi-layered' instruments. However, envelopes, lfos and 'shared' waveforms are not supported in this release. I hope to add those for the next release though I can't promise anything.

Added support for reading and writing Casio FZ-1 dump files (.FZB, .FZF, .FZV) as stored by the programs FZDump or FZCom.

Redesigned the MIDI SDS transfer dialog. Added support for SDS closed loop (handshaking) transfers. This should fix the problems that previously occurred with some synthesizers. Added Midi SDS 'request sample' command.

Larger and better property sheet windows. E.g. a much improved layer keymap editor!

Added many more keyboard shortcuts.

You can now use your favorite waveform editor as a 'helper app' (registered version only). You can use it to edit, record, or otherwise process waveforms in *Awave* without having to manually save and reload it from the disk. You can also have a 'mixer' and an 'on-screen keyboard' helper app that can quickly be launched from a menu.

The right mouse button menu is now extended with an 'add instrument', 'add layer', 'add region', or an 'add waveform' entry (depending on the item type selected).

Added new option 'item name' to the naming schemes for the 'save all in multiple files' command.

Added 'per layer override articulation' to the internal instrument 'architecture' in addition to the 'instrument global' and 'per region override' articulations. Most synthesizers only support articulation at one of these three 'levels'. The problem is that they do not all use the same level, some use per layer, some per region... *Awave* uses this three level structure so that you can always edit at the 'level' used by your synth. There is now also a 'articulation level' conversion feature in the 'process all' box that let's you force articulation to a specific 'level'. It uses the same 'propagate - average' algorithm as used when saving stuff (the FAQ now contains a note that explains that in more detail).

Divided the process all dialog into three pages for 'waveforms', 'instruments' and 'articulation'.

Added support for reverb and chorus level parameters (SBK, SF2 and FFF file formats, though the current GUS PnP drivers do not seem to use these settings from the FFF file…)

Added support for .KRZ multi-layered programs.

Added support for reading Dialogic ADPCM subformat WAV files.

Fixed all known bugs- yes really!(to be more specific, all the following 'bugs' in v4.0 have now been fixed: 12bit Midi SDS data not read properly - win 3.1 open multiple file not working - very unstable after double-clicking a layer in instrument property dialog - written SBK's were incompatible with a few programs - SDS transfers dropped packets for some synths - reading WFD files with ROM samples crashed - only first file read when dragging multiple files from the explorer - canceling a 'merge' with an unsupported file format hanged - failed reading VOX files with text comments - SampleCell II PC export didn't work - incorrectly read a few QuickTime files using 'twos' data format)

✖

version 4.0

A brand new instrument architecture with a *Collection -> Instrument -> Layer -> Region -> Waveform* hierarchy complete with new and updated property sheets. This means: layers, amplitude and pitch envelopes, lfo's, shared waveforms, and more! *Awave* supports a superset of the forthcoming DownLoadable Sounds (DLS) level 1 architecture specification.

Added support for drum kits for .INI+.PATs, .FFF, .SBK, .SF2, .WFD/.WFB formats.
Added (multi-)layer support for .DLS, .EFE, .FFF, .INI(mWave), .SBK, .SF2, .WFx formats
Added envelope support for: .DLS, .PAT, .SBK, .SF2, .WFx formats
Added LFO support for: .DLS, .FFF, .PAT, .SBK, .SF2, .WFx formats

Added *beta* DLS level 1 file format support, import and export (.DLS). The format specification haven't been formally completed yet...
Added support for Sample Cell II, PC and MAC type files import and export (.INS).
Added MPEG layer 3 and MPEG-2 LSF support as well as improved the decoding speed for layer1 and 2 (.MP2/.MP3).
Added support for reading IMA4 format sound from QuickTime movies (.MOV).
Added Talking Technology Incorporated file import, uncompressed only (.VOX).
Added support for very old (v1.00) UltraSound patches (.PAT).
Added raw CCITT G.723 ADPCM (3bit/24kbits or 5bit/40kbits) read and write (.G723).
Added Pinnacle Sample Bank file import support (.PSB). Before you ask: *No*, I will never support exporting to this format. According to TurtleBeach, it will soon be replaced with a 'real' Pinnacle bank file format (.PBF) which I *do* hope to be able to support in a future release.

Added an installation program; SETUP.EXE.

Added a toolbar.

You can now create loops by 'crossfading'; use the 'Crossfade loop' option in the waveform's 'audio processing wizard'

Reworked the software synth code, increased the polyphony in DirectSound mode to 16 simultaneously playing waveforms and added support for multi-layering. Also fixed a number of things that could sometimes make it hang when using it for auditioning.
Fixed a bug that could cause a crash when using an external keyboard for auditioning.

Added loop 'play' button to the loop editor.
Combined the two '*Convert all->Resample*' and '*Convert all->Convert loops*' commands into one '*Process all...*' command and added options to: *Split all stereo* waveforms into mono waveforms and create 'dual layered' instruments for any instruments using those waveforms, *Combine non-colliding layers* in instruments, and: *Split multi layered instruments* into several single layered instruments.
The 'Save in multiple files' command now has a dialog box where you can choose how the output files should be named. And, there is now (after strong demand :-)) an option to use the source file names as the base for the export names.
Added 'Auto property sheet' option which, when turned on, always keeps a property sheet open for the currently selected item in the tree view.
Added 'Delete unused waveforms' command to the Edit menu.
Programs are now numbered 0..127 (not 1..128 as before)
Added copyright parameter for waveforms (.AIF, .WAV). Moved 'Comment' parameter from instrument to collection.
Added option to explicitly enter the 'selection range' in the Audio Processing wizard.
Added display of the length (time) of the selection in the waveform editor.
Added 'Insert silence' option to the audio processing wizard.
Now allows the Midi SDS transmit 'channel' to be in the range 1..64. And now automatically closes down auditioning while receiving Midi SDS in order to avoid conflicting use of midi in device.
Added a -NOMSG command line switch that turns off all error and warning messages.

Fixed problem with losing the 'selection' after closing a property sheet.
Fixed bug when saving .EFE files: all names will now be converted to only uppercase characters, numbers, and -, *.
Fixed bug in the instrument keymap editor (now named the layer keymap editor).
Fixed some problems with WaveFront bank files (.WFB) that made WavePatch dislike *Awave*'s files.
Fixed bug in Bi-directional to Forward loop conversion (the new loop end point was bogus); also made some improvements to the loop point editor.
Hopefully fixed a bug with .IT instrument 'pitch' and updated the .IT code to handle IT v2.0 files.
Fixed bug with fine tuning in .PAT files
Fixed problem with the audio player sometimes stopping prematurely in DirectSound mode.
Fixed bug with Yamaha SY-series wave files.
Changed some shortcut keys to correspond to the order they appear in the toolbar.

✄

version 3.2

You can now explicitly select 'subformat' type when saving to an export format.

Added US Robotics voice modems / QuickLink format (.GSM) import and export.

Added Sound Designed II flattened import and export.,+ resource fork only import (.SD2).
Added QuickTime (.MOV) audio import.
Added GSM subformat to .WAV's export.
Added Fast Tracker instrument (.XI) export.
Added speech data (.SPD) import format.
Added Akai MPC3000 wavefile (.SND) export.
Added µ-law support when writing .FFF files.
Added µ-law support when writing .WF? files.

Can now have multiple property sheets and AudioPlayer's as long as no-one is for the same 'item'.

Added a 'Use Windows multimedia services, unbuffered output' to the AudioPlayer property tab for those that has had problems with no sound in the AudioPlayer.

The Awave AudioPlayer synth has been rename into 'Awave software synth'.
Fixed the worst 'popping' problems in the software synth.
The software synth now supports pitch bend and all notes off midi messages.

On the Edit menu there is now a submenu 'Process all' with two subitems:
- Resample..., resamples all waveform to a given sample rate.
- Convert loops..., converts all loops to common format.

Added support for Microsoft DirectSound (optional, select 'Use Microsoft DirectSound services' in the AudioPlayer property tab).
It's benefits are:
- 		Lower memory usage for the AudioPlayer and the software synth.
- 		Lower latency for the AudioPlayer and the software synth.
- 		Allows multiple simultaneous AudioPlayers as long as they do not play the same item.
- 		The software synth is 8 note polyphonic.

Auditioning GUS & WaveFront: Will load as 16b if it fits in memory, else it will load as 8b if that fits, else it fails...

Improved the "Convert to stereo" selection in the audio processing wizard so that you can now combine two waveforms into one stereo waveform.
Added "Statistics" selection to the audio processing wizard.

Added option to enable and disable association of file formats.

Fixed bug with .EFE files (hanged when editing a waveform on an Ensoniq)
Fixed problem with recognizing .PAT files from 'Howling Dog Systems PowerPatch series' CD-ROMs.

Various minor bug fixes and improvements.

✖

version 3.1

Added GUS PnP .FFF format import/export.
Added Yamaha SY-series waveform (.W01) export.
Added Dialogic adpcm file (.VOX) import & export.
Added Sound Designer I format (.SD) export.

Added DiamondWare Digitized file (.DWD) import & export support.
Added OKI MSM6376 synth chip format (.PCM) import.
Added raw GSM 6.10 audio stream (.GSM) export.
Added Digilink (.DIG) format import.

It's now possible to select multiple files for opening in the file open dialog.

Added 'Midi SDS receive' to Edit menu.
Added 'Send all' to the MIDI SDS transmit dialog. This frees you from having to press the 'Start' button for every waveform when downloading banks or instruments; useful if you want to dump a lot of samples...

The 'AudioPlayer synth' now supports forward looping.

Added 'Convert reverse to forward' loop to audio processing wizard.

The AudioPlayer command line interface now accepts directory names as well as file names, when a directory name is given, all recognized audio files in that directory will be played in turn.
Added -LOOP switch to the AudioPlayer command line interface to repeatedly play the indicated file(s). Break the loop by closing the AudioPlayer with the [x] button. Ex: To repeatedly play all audio files in the current directory, type on the command line: "start AWAVE -AP -ONCE -LOOP ."

Various minor fixes and improvements:
Fixed bug with TB WaveFront cards auditioning that made it require wfgate.dll v1.3.0.6, should now work with other versions as well,
Reading EDE files bug fixed,
Saving TX16W W01 files sometimes crashed, fixed that,
In the instrument keymap editor, you can now click on an 'assigned range' (rather than drag with the mouse over a range) to change that waveform assignment,
Fixed problem with Ensoniq instruments larger than 1MB,
et c...

✁

version 3.0

100 file format barrier broken => I must be crazy :-)

Added auditioning for all digital sound cards! Emulates a very simple monophonic synthesizer (no loop support though...).

Added cut, copy, insert and paste to what is now the waveform 'editor' in the waveform property sheet.
Added Power / Current graph toggle in the waveform properties Editor tab.
Added (time domain) resampling algorithms (nearest neighbor, linear interpolation, sin(x)/x filter) to the Audio processing wizard.
Added Bidder to forward loop conversion to the Audio processing wizard.
Added fade in, fade out, trim, cut, reverse, remove silence, maximize amplitude, set loop point, and smooth out selections to the Audio processing wizard.
Added 'Split into mono channels' command to the right mouse button menu for stereo waveforms.

For non-standard playback rates, the Audio player Always now resamples the waveform to an appropriate standard rate.
Decreased the need for (physical) memory when using the Audio Player. Should reduce the disk

crunching for large files...
Improved the accuracy of the AudioPlayers frequency spectrum graph.
The AudioPlayer position bar in now draggable, both when stopped and when playing.
Added a -NOWIN command line switch to launch the AudioPlayer and play the sound without
displaying the AudioPlayer window.

Tweaked the file open dialog to show detection info on selected files. Also increased the visible length
of file formats combobox list.

Fixed what I hope is the last two bugs in my .SBK writing code :-)- (bugs occurred whenever there
were names longer than 19 chars or when there were 'shared' or 'lonely' waveforms...)
When loading an .SBK which uses hundreds of ROM-samples you no longer have to press OK a
hundred times...
Fixed incompatibility between my .AIF writing code and a few programs.
Fixed bug when writing .WFB's.
Added log/status window when auditioning using the Maui/Tropez.

No longer automatically creates instruments for 'lonely' waveforms when saving bank items. You have
to specifically use the new 'Autocreate instruments' command (on the File menu) before you save if
you want it.

If you want to edit the key assignment for the virtual keyboard you can now do that with the registry
editor (regedit.exe) that comes with Windows 95. Open this_computer->HKEY_CURRENT_USER-
>Software->FMJ-Software->Awave and edit the mtVKeys string. It's just a string of (normally 29, but
maximum 36) uppercase characters holding the ASCII key to assign to a note, beginning with C3 for
the first character and then in increasing order...

On request, I have added a few 'keycuts' (F5 - Save as..., F6 - Open new file... F7 - Play audio... F8 -
Properties...).

Added WAVmaker program files (.PRG) import and export.
Added Yamaha TX16W waveform export (.W01).
Added support for 16bit Roland samplers floppy disk images (.SDK). Now also tries to convert the
instruments for .SDKs (both 12bit and now 16bit) files although it may not always get it right.
Added Digitrakker instrument (.IST) format as well as v1.x module (.MDL) import.
Added Extreme's Tracker (.AMS) format.
Added FunkTracker module format (.FNK).
Added Impulse Tracker modules/instruments/samples (.IT).
Added DisorderTracker2 module (.PLM) and sample (.PLS) files.
Added Poly Tracker module (.PTM).
Added ABT Extended MoDule (.EMD) format.
Added Velvet Studio Modules (.AMS) / Instruments (.AIS) / Samples (.ASE).
Added DigiTrekker module (.DTM).
Added DCM module format (.DCM).
Added Typhoon vOice files (.O01)
Added Typhoon wave files (.C01)
Now reads more Ensoniq synthesizer files (.Efx) and disk images (.EDx, x=A,E,K,Q,S,T,V)
Added raw CCITT G.721 ADPCM data (.g721) read and write.

Added an experimental 'Upload bank ' command, related to the auditioning feature. You can use it to
upload all of the instruments in the bank to the wavesynth. As it is still in the 'experimental stage ', I

can not guarantee that it will work as intended. The idea is that, using this command, you'd be able to develop 'cross-soundcard' compatible instrument bank files. You'd then be able to distribute them with your midi songs and use Awave to upload them to any one of the supported wavetable sound cards. If there is enough interest, I might be able to produce freeware, standalone, 'bank loaders ' for the GUS, AWE32, Maui/Tropez respectively.

Various minor fixes and improvements.

✄

version 2.7

Fixed the SB AWE32 auditioning to work with the new Win95 drivers. The AWE32 auditioning now uses the new .SF2 file format so in fact these latest, .SF2 compatible, drivers are now required (for auditioning with the AWE32). Note! They had only been released in beta (although public beta) versions at the time this was written.

Changed the loop editor to a nice 'dual layer' view.

Added an 'Audio Processor wizard' connected to the 'Process' button in the waveform properties box's waveform tab. Only 16->8bit, 8->16bit, mono->stereo and stereo->mono conversion algorithms are available for the moment.

Added spectrum analyzer option to the Audio Player (uses a Fast Hartley Transform (FHT) algorithm on 256 samples to compute a 128 band power spectrum, all in real time).

Added 'Virtual Keyboard' feature, i.e. when using the auditioning feature (with one of the supported wavetable cards) you can now play on the computer keyboard (if you do not have an external midi device).

Added a 'programs' tab to the bank property sheet where you can easily view (and edit) the instrument to midi program number assignments for the bank.

Can now save whole instruments to Ensoniq .EFE files in addition to single waveforms (i.e. more than one waveform/instrument are now supported).

Added Windows Clipboard import.
Added .AVI import support (audio part only...)
Added ultrasound initialization file (ULTRASND.INI) melodic bank import as well as 'export' (.INI extract + bunch of .PAT files).
Added support for writing μ-law and A-law format .AU and .WAV files.
Added Matlab variables binary file (.MAT) import and export.
Added .VAP annotated speech files import.
Added .UB, .UW, .UDW, .SB, .SW, .SDW, .ULAW, .ALAW import formats and changed .RAW, .SND export formats into .UB, .UW, .SB, .SW, .ULAW, .ALAW. Here U means unsigned, S signed and B, W & DW means byte (8bit), word (16bit) & dword (32bit) respectively.
Added Covox 8 Bit File (.V8) import.
Added Ad Lib Gold Sample (.SMP) import
Added Sonic Foundry Sample Resource (.SFR) import

Fixed AudioPlayer scope bug that made it go faster than the sound on some soundcards (read Sound Blaster when playing 16bit and/or stereo data).
Whenever the selected WaveOut device is busy playing something else, the Audio Player will now

look for, and use, any other waveout device(s) you may have installed.
Fixed SBK/SF2 bug that made it hang sometimes when saving.
Fixed MS ADPCM (.WAV) format decoder bug.
The mandatory small fixes and improvements.

version 2.5

Changed the name from 2PAT into the more appropriate *Awave*. All registered users of 2PAT are, of course, also registered to *Awave*.

Waveforms are now cached in memory. This eats memory but is necessary for some of the new as well as for future editing functions. It also makes auditioning more responsive.

A visual loop point editor page has been added to the waveform property sheet.

A waveform display page has been added to the waveform property sheet (true RMS display, plus peak indication)

Auditioning feedback is now available when editing stuff in the property sheets (i.e. if you have one of the supported wavetable synthesizers). Just hit the Apply button to download the changes to the synth.

An oscilloscope display has been added to the AudioPlayer.

Added option to force the AudioPlayer to use only standard sample playback rates (some buggy drivers says they can play things that they can't!).

Added -ONCE (closes the AudioPlayer when finished playing a waveform) and -DELETE (deletes the source file when finished) command line options in addition to the old -AP (launches in AudioPlayer mode) switch.

Auditioning with the SB AWE32 no longer uses a stand alone program, but now requires the latest drivers with 32bit application support, more specifically, with the AWEMAN32.DLL file.

Added an 'Insert File' command that lets you merge several files.
Added a 'Save All As' command to allow you to easily save all instruments or waveforms.

Added 'MidiSDSDump' virtual export format to allow Midi transfers of waveforms to external synthesizers.
Added Ensoniq EPS instruments, .EFE, export (for single waveforms only).
Added Sound Font v2.0 (.SF2 files) export. CL says they will support SF v2.0 during the 2nd quarter of '96 so in about a years time it will probably have replaced SF v1.0 and .SBK export support will then be dropped from this program.
Added .VOC export format (note! files are written using the new and much more capable 'type 9' subformat (well it's not that new, but there's still a few programs that do not support it)).
Added .S3I, Digiplayer/ST3 sample fileformat export
Added text formatted audio, .TXT export (the old text description format renamed to .DES)
Added MWave DSPs software synthesizers MWSYNTH.INI bank read support as well as .MWS text file export, containing an instrument extract that can be manually pasted into mwsynth.ini.

Added MPEG audio layer I, I and III import, .MP2, .MPA (very slow and memory hungry though).
Added GSM 6.10 audio stream import (.GSM)

Added SoundStage .SFI/.SFD import.
Added Farandole .F3R blocked linear module format (1.0).
Added SBStudio II, .SOU, .PAC sound & package files import.
Added .EPSP audio file import.
Added .NIST SPHERE import.
Added .SPPack sound sample file import.
Added .INRS Telecommunications audio file import.
Added: MAUD sample format import.
Added .SD1 Sound Designer I file import.
Added PSION a-law file import
Added Mus10 audio file import
Added Sound Tools HCOM format import.
Added µ-law and IMA compression support for .AIFC files.
Added CCITT G.721 & G.723 ADPCM as well as IEEE 32bit and 64bit floating point support
for .AU.
Added CCITT G.721 & G.723 ADPCM, GSM 6.10 and MPEG audio support for .WAV.
Added support for the new type 9 .VOC blocks as well a µ-law and A-law compression.

Some corrections/bug fixes for loop points of .KRZ and a few other file types.
Fixed bug with many .GKH files appearing 'empty'. Now also automatically removes the duplicate
but 'noisy' wavesamples that appeared from multi-layered instruments..

Added fast auditioning on/off switch to the Options menu.

Much more transparent file type associating behavior. No longer will you be prompted if you want to
for replace types that have already been associated with other programs. You'll have to do that yourself
if you want to. Registers 'convert' and 'play' command verbs in the new Windows 95 shell extension
fashion (so multiple programs can cooperatively assign different commands for a given type). This
doesn't work very well under Window 3.1 so this feature has been disabled when running under
Windows 3.1. You can of course still manually associate file types using the File Manager.

And, as always, a whole bunch of bug fixes and minor enhancements.

✖

version 2.0

User interface enhanced based on the Windows 95 UI guidelines. Runs under Windows 95, Windows
NT 3.51 or Windows 3.1 + Win32s v1.3.

The really stupid "couldn't register app class" bug fixed (occurred on many Win32s systems).

Now writes converted stuff to disk first when you actually saves or auditions anything, instead of
doing everything in one sweep when opening a file, as was done in previous versions. This change has
made possible a number of the enhancements below.

Tree list view with nodes and leaves for bank file, instrument and waveform levels. You can now save
stuff from any of the 'levels' (Note only a few of the export formats support all levels). There's also
Property sheets for the items of the various levels. Most stuff is editable!

You no longer have to decide on export format before you load a file. Just select from a list in the Save
dialog.

Auditioning now works regardless of what you're going to export to.

Sound Blaster AWE32 auditioning support added.

A virtual '.DAC' export format added. Saving to this format will simply play the item on a digital audio output device. So users that doesn't have a GUS, Maui or AWE32 can now play sounds as well, provided you have some sort of basic audio card. An 'Audio Player' item in the right mouse button menu acts as a shortcut for saving to this format.
Audio Player mode and -AP command line switch added. Lets 2PAT be used as a multiformat digital audio player. Great as a helper app for your favorite Web-browser.

A virtual '.CLIP' export format added. Saving to this format will simply copy a wavesample to the Windows clipboard.

.AU export format added.
.IFF export format added.
.SBK export format added as well as improved import support..

Smart file format recognition, Tries to recognize the format of a file correctly, even if it has no, or even an incorrect, file type extension.

FastTracker 2 .XM (module) and .XI (instrument) files read supported.
Farandoyle .FAR tracker modules read support.
Oktalyzer .OKT tracker module read support.
Sndtool sound file, .SNDT read support.
Sounder sound file, .SNDR read support.
IRCAM SoundFile format, .SF read support.
Raw Audio-CD format data, .CDR read support
MikMod 'UniMod' format, .UNI read support
Digital Sound Module tracker format, .DSM read support
Text formatted audio data, .TXT read support
Delusion/XTracker Digital Music/Sample File format, .DMF, .DSF read support
Digitrakker Module/Sample File, .MDL, .SPL read support.
Signed 8bit Sample data, .SAM read support.
Farandoyle Linear module (2.0) format, .F2R read support.
Macintosh SoundCap/SoundEdit recorded instrument format, .DEWF read support.
Protracker Studio Module Format, -PSM read support.
OctaMED tracker modules (MMD0|MMD1|MMD2), .MED read support.
Machintosh Audio Visual Research sound files, .AVR read support.
Ensoniq, .EUI disk images read support, somewhat like .GKH, which now also reads subdirectories.
US telephony audio format, .UL read support.
AKAI MPC3000 sample files, .SND, read support

Stereo support for a number of formats (WAV,SND,RAW,S3M,S3I,IFF,AU,AIFF)
A-law and μ-law encoded .AU files now supported.
A-law, μ-law, MS ADPCM and DVI/IMA ADPCM encoded .WAV files now supported (read).
16bit 'SV16' flavor of IFF files now supported (read/write).
A bunch of additional .MOD flavors now recognized.
Added cue-points support for looping in .WAV files (read/write).
Fixed ugly bug with .SDS import and export.
Reads and writes 'inst' chunks in .WAV files.

A 'Dummy' midi in device now added so people without a midi-keyboard can use the audition feature of 2PAT by using any 'on-screen keyboard' program.

Added 'Uninstall' button to the program properties box that can be used to clean up the registry.

Imposed a somewhat annoying save times limit. Registered users receive a personal keycode that removes this.

✖

version 1.5

Reworked user interface for Windows 95. 'Right mouse button' based, uses the registry, associates file types, uses long filenames, new icons, resizable window frame et c. No longer uses CTL3D32.DLL since that's no longer necessary for the 3D look. Will still run under Windows 3.11 with Win32s v1.1 and up but will probably not look very good; Win32s v1.3 will probably fix that when it's out.

Instrument info box with info corresponding to a .DES file.

Improved Kurzweil *KRZ* support; fixed the sample rate and a few other things. Now creates instruments from both the 'sample' and the 'keymap' level, since the 'keymap, i.e. instrument, level doesn't always work....

Added *AIF* as export format.
Added *SND* as export format.
Added *RAW* as export format.

Fixed problem with not being able to launch WinHelp from arbitrary directories.
Assigns unique names for 'unnamed' samples and instruments (fixes problem with WaveFront based cards).
Fixed problem with 'unnamed' instruments in .PAT files.

✖

version 1.4

Improved support for Turtle Beach WaveFront based cards! (Maui / Rio / Monterey);
Reads .WavePatch's.WFB, WFP and WFP files (v1.2 or later only).
Exports .WFP files.
Supports downloading instruments to the Maui card, using the WF GateKeeper (WFGATE.DLL).

Reads ScreamTracker 3 instruments (S3I).

✖

version 1.3

Now a Win32s (v1.1 or later) program.
CTL3D32.DLL replaces CTL3DV2.DLL.
FMJMTHRU.DLL no longer needed.
Saves configuration in the Win32 registry instead of in an .INI file.

Now reads the GKH disk image files in addition to INS/EFE for Ensoniq EPS.

Bare bone .SBK import support (EMU SoundFonts Banks / SB AWE32); at wavesample level.

✖

version 1.2

Restructured for multiple export formats, currently:
- Ultrasound patches (.PAT)
- Turtle Beach Maui (.WAV+.WFS+.DES)
- Ms Windows wave files (.WAV+-DES)
- Midi Sample Dump Standard files (.SDS+.DES)

Config dialog window where you can choose export type and midi in device.
Saves config in 2PAT.INI.

You can start 2PAT with a filename on the command line and that file will be loaded at once; so now you can associate various file extensions with 2PAT in the File Manager and launch 2PAT by double clicking on a file.

Reads and writes .WAV files with the (new Ms standard) smpl (root key, fine-tune, loop points) and ICMT (name) chunks.

✖

version 1.1

Uses Microsoft's 3D controls extensions, CTL3DV2.DLL instead of Borland's BWCC.DLL as earlier.

Improved Ensoniq EPS support (.INS, .EFE, .GKH).
Now treats the sample rate correctly as well a doing a coarse decay rate conversion.

Now supports fine-tune for a few formats.

Better at cleaning up temporary files after itself...

✖

version 1.0

Reads: .AU, .INS, .EFE, .VOC, .SMP, .PAT, .SDK, .SDS, .SDX, .SYW, .TXW, .IFF, .AIF, .UWF, .MTM, .ULT, .WAV, .SND, .FSM, .669, .KRZ, .STM, .S3M, .MOD (M.K., 4CHN, 6CHN, 8CHN, FLT4 or FLT8) and raw files

Writes: .PAT and .WAV.

Now a Windows hosted program (Win16). And the manual is in WinHelp Hypertext format.

Lots of improvements over previous beta versions…

✖

version 0.1ß

Nothing fancy, reads 4ch MOD's, writes PAT's, ULTRASND.INI extract.

✖

Frequently Asked Questions

✖

Sections

✖

F.A.Q. – Shareware and registering

✈

Q   How do I register *Awave,* how much does it cost, and how can I pay for it?
A   Please consult the REGISTER.TXT file that comes with the program. It contains detailed
instructions. You can pay by credit card, by check, by bank transfer, or by cash.

Q   Why this &¤#'% nag box in the unregistered version?
A   Early versions did not have it since I though that users should get every opportunity to properly
evaluate 100% of the program. But I soon realized that that wasn't working and that I had to add an
'incentive' to make people register the program. I am sure that *you* are very honest and register *all*
your Shareware programs that you use, even if they do not contain any nag boxes or are crippled in
any way. However, in my experience, 99% of people are not that honest :-). And the nag box do seem
to do it's work; even though I do not like it very much myself...

Q   I downloaded this program some time ago but haven't really had time to try it until now. But now it
says that the '30 day trial period' has expired whenever I try to run it. I still want to try it, what can I
do?
A   Well, I am not entirely sure if I really ought to answer this question, but I fancy myself a kind
person :-) :-), so here we go: Uninstall the program and then reinstall it and you will auto-magically
get a new 30 day trial period. Do not miss-use this! You are still only legally allowed a 30-day trial
period for each new version of this software.

*The following selected **Q**&A's have been reproduced from the Shareware FAQ by kind permission of*
Mitchell Friedman, ASP Public Relations. The Shareware FAQ is copyright 1995-1997, Association of
Shareware Professionals, All Rights Reserved.

Q   What is shareware?
A   Shareware is a marketing method, not a type of software. Unlike software marketed through normal
retail channels, where you are forced to pay for the product before you've even seen it, the shareware
marketing method lets you try a program for a period of time before you buy it. Since you've tried a
shareware program, you know whether it will meet your needs before you pay for it. Shareware
programs are just like programs you find in major stores, catalogs, and other places where people
purchase software -- except you get to use them, on your own computer, before paying for them.

Q   What happens if I like a shareware program?
A   You pay for it at the end of a trial period (typically 30 days) by sending the author a fee he or she has
established for the program.

Q   Why should I pay for and register a shareware program?
A   The same reason you should pay for any program: because it is the honest thing to do. Shareware is
commercial software, fully protected by copyright laws. Like other business owners, shareware
authors expect to earn money for making their programs available. Paying for and registering a
program also entitles you to support from the author and other benefits, as specified by the author.
Moreover, the more consumers who pay an author to use a program, the more likely the author will
continue to improve it and to offer new programs.

Q   How do shareware programs compare with other kinds of software?
A   Consumers who purchase shareware programs receive a level of product support that exceeds what

traditional software manufacturers deliver. Shareware users who need support often speak directly to the actual developer of the program, who is intimately familiar with how it operates and therefore can provide excellent technical support. Shareware authors often fix bugs in programs and add features quickly, based on feedback from users. There is a wide price range for shareware, as there is with software distributed through other channels. In general, many shareware programs cost less than other kinds of software, while some programs cost about the same as retail counterparts.

Q What do I receive when I pay for a shareware program, besides the use of the program?

A Typically, the same things you receive when you pay for other software: support by telephone, fax, computer bulletin board, and/or through online services such as America Online, CompuServe, and Microsoft Network. Many authors also send manuals, reference cards, and other printed materials, and may offer free upgrades. Every shareware program is different, so the version you purchase comes with different materials. Documentation files included with the program describe the benefits you receive by paying for and registering a particular shareware program.

Q What happens if I don't like a shareware program?

A You simply stop using the program, and remove it from your system. Since you have had the opportunity to try the program first before paying for it, you lose only the tiny amount of money you spent to download the program or to acquire it from a vendor or other source.

Q I ordered a shareware program from a catalog and paid for the disk. Why should I pay more now?

A Shareware vendors distribute shareware versions of programs, charging a small fee for the costs of disk duplication and advertising, plus a small profit. Most shareware authors allow this type of distribution so you'll have a chance to try their programs. However, none of the money paid to a shareware vendor goes to the author. If you try a shareware program, then continue to use it after the trial period, you must pay for and register the program. The same principal applies if you buy a shareware disk at a computer show or find a shareware program on a CD-ROM disc or at a store.

✖

F.A.Q. - Converting issues

✖

Q   Can I convert a .WAV (or any other digitized audio) file to a .MID?

A   This is the one most common question that I am asked. The answer is: NO WAY! I don't have enough 'space' to explain this in detail here but suffice to say that it is a very difficult problem since it is two fundamentally different data types (e.g. see http://www.lysator.liu.se/~zap/tutorial/formats.html). There is no computer program that can do it in a very 'reliable' way (the best I have heard of is software that can recognize the note of a single instrument played monophonically). You really need a highly trained professional (human) musician to do it. In this case the human brain beats any computer.

Q   Why can't I convert a .MID file and to a .WAV (or any other digitized audio) file?

A   This is the second most common question that I am asked. A .MID, or 'midi file' contains the 'score'; instructions on how a synthesizer should play a piece of music. *Awave* only handles 'wavetable' audio (see previous Q) which is a different thing altogether. Of course, I could do a 'software synthesizer' that could 'play' a .mid file and output it as a .WAV file rather than to the speakers. However that would be a lot of work to implement, and now I really mean *a lot*, especially if you want professional quality results. Also it would require a midi instrument library which would have made *Awave* into, say a 10MB package :-). Besides, it falls sort of outside what *Awave* is intended to be.

Q   Why can't it convert a whole .MOD, .X3M, whatever, song into a .WAV?

A   See the previous Q. It can only convert the 'instruments' used in these files into other formats. 'Instruments 'consists of various waveforms or 'recorded sounds'. These are then used in a 'song' just like a .MID file.

Q   Why can't it read the Adlib's .ROL (or similar) format(s)?

A   *Awave* can only handle 'wavetable' based audio and instruments, i.e. things based on digital recordings of actual sounds. The adlib (and the .ROL files) uses FM-synthesis which is a 'mathematical formula' for producing sounds and simply isn't compatible with *Awave*, nor with any of *Awave's* 'export formats'.

Q   I can't save in *whatever* format even though you seem to claim that *Awave* supports this format!

A   Not all formats can store *all* types of data. Ex: only a few formats support saving whole bank files while most formats (supported by *Awave*) supports saving single waveforms. What type of data you are trying to save is determined by what kind of 'node' or 'item' in the tree view list you have selected. See 'File Formats' in the program help file and look at the export formats list for info about what formats support what data types.

Q   Is there a batch convert function?

A   Yes. You can open multiple files (in the standard Windows fashion, hold down Ctrl or Shift while selecting files) and then save all items by using the 'Save all in multiple files...' function on the File menu. The 'Merge with another file…' function may also come in handy.

Q   Why is there three levels (region, layer, instrument) of   articulation data? What happens with this when I save a file?

A   Articulation consists of 'modifiers' that change the sound over time, e.g. lfos and envelopes. *Awave* supports articulation parameters stored at 'three levels', instrument global, layer override, and region override articulation. Each level can be enabled or disabled. The relation between the three levels

should be interpreted as follows: When a waveform is played, the 'per region override' articulation parameters are used if enabled. If not then the 'per layer override' articulation parameters are used if enabled. If not, then the 'instrument global' articulation parameters are used if enabled. If not, then reasonable, neutral default values are used.

Now, various synthesizers generally only support articulation parameters stored at one of these three levels. When reading files this is naturally not a problem. When writing to a format that supports per region articulation then the process for determining the articulation for a waveform outlined above is used to find the per region parameters to write. But what happens when the output formats only support per layer or instrument global articulation? The approach used is to average the articulation for all the waveforms used in the layer or the instrument, i.e. the process described earlier is used to 'propagate down' articulation to every waveform. All the 'waveform articulations' are then 'averaged upwards' to the articulation level that we want. *Sounds complicated?* This is the best process I have been able to figure out for converting between the formats of different synthesizers. As an example consider if only per layer articulation is enabled in a single layered, two region instrument and we want to save this instrument to a file format that supports only per layer articulation. This process will save the instruments per layer data in the file's per layer data as is natural (although it has first been propagated down to each of the two regions and then averaged back to form the original per layer articulation again). Let's say you then enable the articulation for one of the two regions. When next you save the instrument again, the average of the per layer data (propagated down to the waveform with no per region articulation) and the new per region data (for the articulated regions waveform) is saved as the files per layer data…

Hint! While very useful for making good conversions, this three level articulation structure may only be confusing when editing instruments targeted for a specific synthesizer. In this case you can use the 'edit->process all->articulation' feature to 'convert' all the articulation data to the single level supported by your target synthesizer. E.g. if you will be using the Turtle Beach Maui synth then you can use this to get all the articulation at the 'per layer' level.

Q It does not seem to convert envelope/lfo/layer data for my instrument file format!

A There can several reasons:

a) Your file format do not support that particular kind of data. See the table below.

b) *Awave* do not support that particular feature of that particular file format. This can be because: i) it uses a format completely incompatible with the one used by *Awave* (this can often be the case with envelopes, of which there seems to be as many varieties as there are synthesizers...), or ii) it is an obscure file format for which I have not cared to implement every finest detail. See the table below.

c) Perhaps it is there but has been 'averaged' out so much that you don't hear it. This can e.g. happen if you have an instrument with 10 waveforms, each with their own per region articulation, and save it to a format that only supports e.g. instrument global articulation. If one of the regions have articulation differing from the others then it still only affects 1/10th of the instrument global articulation data that is saved which may be too little to hear. Read the previous Q for a more in depth discussion about how articulation is treated. To avoid confusion, you can use the 'edit->process all->articulation' feature to set all articulation data to the single level supported the output file format. See the table below for what level articulation various file formats supports.

Here follows a table of what is implemented for some of the most important instrument file formats:

Format	*Layers*	*AmpE*	*PitchE*	*LFO*	*Articulation*
.DLS+	-	Y	Y	Y	'per instrument' (melodic) / 'per region' (drum kits)
.DLS*	Y	Y	Y	Y	'per region'
.EFE	Y	N	N	N	'per region'
.FFF	Y	Y	N	Y	'per region (envelopes) / 'per layer' (lfos)
.PAT	-	Y	-	Y	'per region'
.PBF	Y	N	N	N	'per layer'

.SBK	Y	Y	Y	Y	'per region'
.SF2	Y	Y	Y	Y	'per region'
.WFP/B	Y	Y	Y	Y	'per layer', max 4 layers.

Y Supported by *Awave*,
N Supported in some sense by the file format, but *not* by *Awave*,
- Not supported by the file format.
+ Conforming to the DLS level 1 standard.
* Extended DLS level 1 - FMJ-Software' proprietary (but forward compatible) extension while waiting for DLS level 2...

Q Please, can you add Roland .SDK or Kurzweil .KRZ file export?
A Sorry, the format of those files have never been officially released by Roland or Kurzweil resp. They seem to regard them as company 'secrets'. What I know, stems from what various people, not associated with Roland or Kurzweil, has been able to find out through inspection and experimentation and then let on to me. It is just enough to enable me to read those files but I can not write 100% correct files.

Q Can I read Kurzweil/Akai/Roland CD-ROMs?
A No, they do not use any standard (e.g. ISO-9660) file format so your CD-ROM/Windows can't read them.

Q The converted instrument looks all right in *Awave*, then I upload it to my synth but then I can't hear anything when I try to play it!
A There can be several causes.
1) Make sure that you have in fact uploaded it correctly to your synthesizer correctly. As this is done differently for almost every synth, I can't help you there, you must know what you are doing.
2) Make sure that you have assigned it to the correct program number and midi channel as you are trying to play it 'on' and check that all midi routing et c are OK so that midi data actually gets where you want it to.
3) Look at each 'region' in the instrument; the region(s) is used to trigger a waveform. Look at the key range parameter. The waveform is only triggered if the midi key lies within the key range (e.g. drum sounds is often only triggered by one single key).
4) Look at each 'region' in the instrument. Look at the velocity range parameter. The waveform is only triggered if the midi key down velocity lies within the velocity range.
5) Look at both all the regions and all the waveforms. Check the 'damping' parameter. This adjusts the 'volume' at which the waveform(s) is played. Perhaps it is set so low that you can't hear anything?

Q Why does loops sometimes 'click and pop' when converted to my synth's format?
A A possible explanation may be that there are reverse or bi-directional loops and your synth may not support those. In these cases those loops are saved flagged as normal forward loops but they are not actually converted to forward loops. However you can do that manually by: Selecting 'Edit->Process all'. Click 'Convert all loops to common formats'. Click 'Finish'. Then save it in your synth's file format.

Q When saving stereo data to a format that only support mono waveforms (like .PAT) or when doing an SDS transfer, how do I specify which channel of the stereo data should be used?
A In these situation *Awave* always automatically merges the channels by an 'arithmetic average' algorithm. However, you can use the waveform item->Properties->Editor->Process wizard to specifically convert the data to mono before you save it; there you can choose between 'average of channels', 'left channel only' or 'right channel only' algorithms.

Q For some synthesizers formats (like Ensoniq & Roland formats), *Awave* can only read 'disk images', why can't it read the disks directly?

A Because they are not in FAT compatible format and thus are not readable by the OS. This   means that it   would require some serious low-level reprogramming of the floppy disk controller in order to read them. While this is possible under DOS and perhaps under Win3.1, it is far more difficult under Win95, not to speak of   under Win   NT, requiring (three different) custom ring 0 device drivers (for the three resp. OS's).This is not a project I care to embark upon... There are   Shareware/Freeware DOS programs for the resp. synthesizers   that you can use to make disk images instead.

Q I'd like to be able to add a few audio file format conversion functions to my commercial software, is it possible to license parts of the *Awave* code from you?

A Yes it is, for more information look at my www page, URL: "http://hem.passagen.se/fmj/awlcpack.html"

Q I just hangs when I try to read a 'raw 32 or 64bit floating point' file!

A This may happen if it isn't in fact a floating point file, it is a corrupted file, it is not in IEEE format or something like that...

Q Why can't it read .CDA (i.e. Audio CD's under Windows 95)?

A I have not been able to figure out how / if   this could be done under Windows 95; nor has anyone else to   the best of my knowledge. It is possible to do under DOS/Windows 3.1 though (by calling the MSCDEX driver which isn't used under Windows 95 or Windows NT). *Awave* has to run under all Windows platforms so I can't do that... There are a few Shareware programs around that should do the trick (they generally run under DOS). Beware that some CD-ROM drives has hardware 'locks' that will prevent them from reading raw data from audio CD's so it still may not work...

Q How do I read a file as 'raw audio' data, i.e. how do I override *Awave*?

A If the file in question is 'autodetected' with '100% certaincy' (in the lower right portion of the file open dialog) as a supported file format then you cant. You can however often go 'waveform->properties->source' tab and change the data type there. This will not work for some types. If the file type can't be detected at all (0% certaincy)then it will be read as 'raw data' and you will automatically get to this tab when you load the file. If however it is detected with say '33% certaincy' and you get an error message when you try to   open the file then you may try to remove the file extension (e.g. turn 'foo.bar' into 'foo') in order to fool *Awave* into not detecting it at all (0% certaincy) and read it as raw data.

Q Why is the first letter in item names sometimes bogus?

A *Awave* attempts to ensure that all items have unique names(since this is required by several synthesizers), this is often   done by 'increasing' the first letter.

Q I'm saving FFF files, why can't I hear any reverb, chorus, or lfo on the instruments in this file even though I have set these parameters?

A For reverb and chorus: Ask Gravis! I do write these parameters to the files, but it seems like the current drivers ignore these 'default effect' parameters. I believe you can insert some kind of midi command in your midi songs to explicitly set reverb and chorus levels instead.
For lfo: The FFF format only supports lfo 'per layer' so any 'per region' or 'instrument global' lfo articulation parameters are averaged over the layers. If you use a multi-region layer where only one of the regions has lfo set to max and the rest of them has it set to 0 then, naturally, the average of them all (that is written as the layer lfo) may be too small to hear…

Q I converted a file to .AU format for use in a Java applet but it doesn't work!

A The standard Java code is really stupid in that it is *very* particular as to what 'flavor' of .AU files it will accept... It must be a 8kHz, 'mu-law subformat' .AU file. First use the audio processor wizard or use the process all feature to resample the waveform to 8000Hz. Then save it as an .AU file and specify mu-law in the subformat list of the save file dialog.

F.A.Q. - Editing issues

Q I do not understand what all these parameters mean, help!
A Please refer to the DownLoadable Sounds Device Architecture - Level 1 for a complete description. unfortunately this document had not yet been released when this was written. When it is, I will place a copy on my www site!

Q My synth doesn't support bi-directional or reverse looping, what can I do about waveforms that uses it?
A Try: 'Edit->Process all->Convert loops'. This will convert all loops defined at the 'waveform' level. Any region 'loop overrides' will be left as is and you will have to sort those out by hand if necessary…

Q Where is all the audio processing functions listed in the help file for the Audio processing wizard? I only see a few of them!
A Some of them are only applicable to some sorts of data (like 'convert mono to stereo' is only meaningful for mono data) while other is only implemented for some data types (like remove silence only works on mono data) and *finally*, many of the functions requires you to select a range in the waveform first. To do that, in the waveform properties editor tab, press down a mouse button over the waveform, drag with the mouse and release the button. To deselect all, just click on the waveform again.

Q How do I do to 'hear' the loop in the loop point editor?
A There are two ways:
1) Press the *play |> loop* button to start/stop playing the loop using the *Awave* software synthesizer. While it is playing, you can directly hear changes while you edit the loop points. There is a delay of second or two, due to the buffering in the software synthesizer...
2) Using auditioning. First enable auditioning. Back in the loop point editor, press the 'Apply' button to update the loop point. Then play it on your keyboard! If you don't have an external midi keyboard and are using *Awave*'s 'virtual keyboard', then you'll have to use some other 'virtual keyboard' program. The one ine *Awave*'s doesn't work in the property sheet boxes (you need the keyboard for typing in numbers there!).

Q How do I create an appropriate amplitude envelope for a 'one-shot' waveforms which really should have no amplitude modification and be played until the end of sample data?
A The simple ADSR envelope provide by *Awave* can't really handle this case. The '*tweak solution*' is to set the attack duration to 0 and sustain and release values to maximum. If the sustain and release values are at their maximum then *Awave* will make sure that it is saved with an envelope appropriate for such 'one-shot' waveforms

Q In the waveform source properties page, why is the data type controls always 'grayed out' for some file formats (like .WAV)?
A These controls are only available for file formats that internally in *Awave* is handled by a 'generic' waveform data reader. Some file formats has more specialized readers that won't let you change these things. There's really no reason why you should change these things in those cases; the data type controls is intended for use when reading raw data formats which lack any 'header information' and you may have to manually adjust the data type. Another reason they may be 'grayed' is if you have done any editing or processing on the waveform data. In that case they are grayed to prevent the

'editing' from being lost (by rereading the data from disk which is what happens if you change the data type).

Q How can I use a stereo waveform in an instrument for my synthesizer?

A If the synth supports multi-layering, you can split the stereo waveform into two mono waveforms (e.g. using the audio processing wizard), pan them full left resp. full right and put them in separate layers in the instrument. This whole process can be fully automated by using the 'Edit->Process all..->Split all stereo waveforms...' feature!
There *is* one catch though. To make stereo samples sound any good you really *must* ensure that the left and right waveforms are played back synchronized. The present version of *Awave* does attempt to tackle this issue as it is handled somewhat differently by different sound cards and I have not yet resolved how to handle it consistently for all cards. Thus you will have to manually 'post process' stereo instruments created by *Awave* Use an instrument editor designed specifically for your sound card and set some flags or such to tell it what and how waveforms should be synched together.

Q What about drum kits? Are they supported?

A Drum kits are treated as just another multi-waveform instrument, but has a 'Drum kit flag' set in the instrument property sheet. Note that different synthesizers select and number drum kits a bit differently; some select them by program number alone, some by bank number, and some by both. *Awave* uses both the bank and the program number as one 'linear' number so that drum kits no 0 to 127 are all in bank 0. Kit 128..255 are in bank 1 et c. This is then translated to or from the numbering scheme used by each particular synthesizer and file format. For maximum portability, it is recommended that you always set the bank number to 0 for drum kits.

Q How do I record a new waveform?

A Click File->Add new waveform. A new 'dummy waveform' containing a few silent wavesamples is created. Select it and click 'right mouse button'->'Run editor app' to launch a waveform editor helper application. This is by default the standard Windows 95/NT4 SndRec32.Exe but you can change it to another program in the Options->Program setup...->Helper apps dialog. The dummy waveform is not opened in the editor application. Now simply record over the dummy waveform, save it and close the editor. The dummy waveform is now replaced with the newly recorded data.

✈

F.A.Q. - Other issues

Q How do I pronounce *Awave* and does it mean something?
A Like *Ae-wae-v*. No, I was thinking along terms like AudioWaves workshop, or AnyWave, or something like that, but I couldn't decide :-)

Q How do I uninstall *Awave*?
A Run *Awave*, then Options->Program setup…->Uninstall. Or, under Windows 95, you can use the Windows 95 Control Panel's Install/Uninstall feature.

Q I want to use *Awave* but I don't want it to associate itself with all those file types, what can I do?
A Go to Options->Program setup…->General and uncheck 'register 'play audio' & 'convert' commands.

Q Where can I find free audio samples and instruments?
A The Internet is good source. Here is a (by no means complete) list of sites you may want to check out:

AWE32 SoundFonts Gallery	http://www.dragonfire.net/~huijia/soundfonts.html
Atmosfear's Synth Gallery	http://www.servtech.com/staff/spetry/
An SB32/AWE32/AWE64 page	http://www.telebyte.nl/sound/awe32.htm
Compact light Studios	http://www.planete.net/~gmeister/
iMEDIA	http://www.lfo.com/
Some AWE-32 files	ftp://ftp.inc.net/pub/awe32/
More AWE32 files	http://www.geocities.com/SiliconValley/park/3452/jns.html
Yamaha TX16W site	ftp://ftp.t0.or.at/pub/sound/tx16w/
Various .WAV files	ftp://ftp.funet.fi/pub/sounds/wav/samples
Roland files	ftp://lotus.uwaterloo.ca/pub/sgroup/samples
Ensoniq files	ftp://oak.oakland.edu/pub/eps/samples
Official Ensoniq site	http://www.ensoniq.com/
The Kurzweil Launch Pad	http://home.sprynet.com/sprynet/palermo/k_main.htm
Some Kurzweil K2000 files	ftp://ftp.uwp.edu/pub/music/lists/kurzweil/sounds
More Kurzweil K2000 files	ftp://bach.nevada.edu/pub/K2000/sounds
Gravis UltraSound archive	ftp://wuarchive.wustl.edu/packages/ultrasound/sound/
Midi SDS format samples	ftp://alf.uib.no/pub/midi/sds/samples
Some AKAI sounds	ftp://ftp.mfi.com/pub/keyboard/AKAI/

Q Can I just use any 'sound data' that I happen to find for whatever purposes that I like?
A No, you must first make sure that it is in the public domain and that it is not copyrighted material. If it is copyrighted, then you will need to have a permission from the copyright owner. Most material that you find on the Internet is public domain though ('unless otherwise stated') but it is up to you to make sure that it really is.

Q What is the program *2PAT* and how does it relate to *Awave*?
A *2PAT* is simply a very old version of *Awave*. There is no reason why you should use it since *Awave* does all it ever did and much, much more! If you find it on any www site or similar, please ask the www site maintainer to replace it with the latest version of *Awave*!

Q I try to load, audition or play this really huge file, but it takes a very, very long time or the program just exits with an out-of-memory message...

A   *Awave* is currently limited to working only with files it can fit into memory. To play huge files you may need a huge amount of *physical* RAM.

Q   I just get a 'Windows error: 32' message when I try to start *Awave*!
A   Look in your AutoExec.bat and Config.sys files to see if 'Share.Exe' is being loaded. If it is, then remove it.

Q   When I start the program, I first get the 'nag' screen, wait 8 seconds, press Continue and then it just dissapears and nothing more happens!
A   One possible explanation for this is if AWAVE.EXE has been tampered with in which case it will promptly exit at this stage. Perhaphs it was corrupted during a file transfer or perhaphs your computer is infected with a virus. Try running an anti-virus program to see if your computer is 'clean' and if it is, then download *Awave* directly from my official web site.

Q   Why don't you make a Macintosh version?
A   With SoftWindows 95 you can in fact run *Awave* on a Power Mac. For a 'native' version, see the next Q.

Q   Why don't you make a DOS/OS2/Linux/Unix/Amiga/Atari version?
A   Because I don't have the time or the motivation. If there's enough people interested (read "I'd register if only you'd do an…"), I might be persuaded to do a command line version, stripped of the user interface and all editing functions...

Q   Why don't you do a French/Dutch/Japanese/whatever language version?
A   Partly because I do not have the necessary language skills. But mostly because developing and maintaining several different language versions would mean a huge amount of additional work and I believe that 99.99% of the users knows enough of the English language to use the present version. If anyone would like to translate the help file, then I'd be happy to supply the necessary English source texts files.

✖

F.A.Q. - Auditioning/AudioPlayer issues

✈

a) General auditioning issues:

Q   The AudioPlayer doesn't work. I get no sound, or it sounds really strange..
A   Try selecting 'Use Window multimedia services, unbuffered output' in the Options->Program->AudioPlayer property tab.

Q   I try to play a .MOD file but I can only hear sound 'snippets', why?
A   A .Mod file contains basically a bunch of wavetable based instruments (like a bunch of .WAV files) plus one 'score' (like a .MID file). *Awave* only handles the wavetable instruments (the 'samples'). Get a program like 'Mod4Win' if you want to play the 'score'.

Q   The virtual keyboards key layout   is not well suited for my non-querty keyboard (e.g. German keyboards. may have X and Y swapped). What can I do?
A   You can edit the key assignment for the virtual keyboard with the registry editor (regedit.exe) that comes with Windows 95. Open "this_computer->HKEY_CURRENT_USER->Software->FMJ-Software->Awave" and edit the mtVKeys string.   It's just a string of (normally 29, but maximum 36) characters holding the ascii key to assign to a note, beginning with C3 for the first character, then in increasing order.

Q   Why doesn't the virtual keyboard work in the property sheets?
A   Because there you may need your keyboard to type in numbers and stuff. Use some 'on screen' virtual midi keyboard program instead (like the vpiano.exe program that comes with Cakewalk Audio).

Q   Why can't I select 'Use Microsoft DirectSound services'?
A   You need to install DirectX v2 or later (available for Windows 95 or NT).

Q   I get a 'DS::CreateSoundBuffer()'message when I set the AudioPlayer to 'Use Microsoft DirectSound services'. Why?
A   You probably have DirectX v1 installed. You need DirectX v2 or later.

b) The Awave software synth:

Q   How can I make the software synth sound much better?
A   Install Msft DirectX (v2 or later) on your computer. Go options->program setup->AudioPlayer and select '*Use Microsoft DirectSound services*'.

Q   When I have the AudioPlayer in DirectSound mode and have enabled auditioning using the *Awave* software synth. Now I can't use the audio card in any other program. Why?
A   With this particular setup, *Awave* keeps the digital audio device open (almost) all the time which prevents other programs from using it. If I don't do it like that, then there would be a 'click' when you start and stop playing a note on the software synth which I think would be worse. You can quickly toggle auditioning on/off using the Options>Auditioning menu item. When it is off, then you can run that other program.

Q It doesn't play reverse and bi-directional loops correctly!

A Yes, they ain't supported by the software synth. They will be played as normal forward loops. Use '*Edit->Process all*' to convert all loops into forward loops.

Q There are 'breaks' in some loops! Why?

A If you have the AudioPlayer set to use any of the '*Use Windows Multimedia services*' modes, and the loop is very short, then the software synth may 'skip a few beats' every now and then during the loop... The solution to this problem is to set the AudioPlayer to the DirectSound mode.

c) Creative Labs EMU8000 synths:

Q Program complains about not being able to load AWEMAN32.DLL when I try to use auditioning on the SB AWE32 (or compatible).

A You must have this .DLL installed in your WINDOWS or WINDOWS\SYSTEM directory. You must also have the very latest drivers that supports the new .SF2 file format.

Q After having used *Awave* and auditioning I sometimes cannot play back midi files properly! Instruments are silent or messed up!

A You need to restore your original midi instrument setup. Quit *Awave* then launch the Awe Control Panel. Use it to clear all user banks and to reload your GM or GS bank.

Q I get this message: AWEMAN32::AWE_ERR_DEVICE_DRV_INVALID

A You may have 'mixed' up driver versions. This happened to some people who had installed the 'beta SF2 drivers' and then an sbw95up.exe package which contained newer SB16 drivers but in fact older AWE32 drivers. Get and install the very latest drivers.

Q After running *Awave* and using auditioning for the AWE32, I have trouble running a few 16bit programs that uses the AWE32. Why?

A This is a bug in early 'beta SF2' drivers. Get the newest release drivers.

Q So, where can I find the latest AWE32 drivers then?

A On the Internet, from: "http://www.creaf.com"

Q When auditioning waveform items, I can only hear anything within a certain keyrange.

A It seems like the EMU8000 synthesizer use by the AWE32 can only interpolate within a certain range so if the note is too far off from the Root key, then nothing will sound.

d) Turtle Beach WaveFront synths:

Q Program complains about not being able to load WFGATE.DLL when I try to use auditioning on the Maui/Tropez.

A You must have this .DLL installed in your WINDOWS or WINDOWS\SYSTEM directory. It is available   from Turtle Beach and it comes (and is automatically installed) with the latest versions of the Windows drivers as well as their WavePatch software.

Q When having auditioning enabled, why does it take such a long time to start up?

A Not my fault. It's the same thing as when starting TB's WavePatch.

Q When I load a bank file (.WFB) in WavePatch to my Tropez or Rio, I get a 'bank not saved for the

Tropez/Rio' message. Why?

A   *Awave* sets 'Maui' as synth name in the WFB files but that is   the only thing that would differ from a Tropez or a Rio bank. Just click OK to load it anyway.

Q   What is this AW2WF16.EXE file?

A   TB provides sample downloading support through something called the GateKeeper which is implemented by the WFGATE.DLL file. But, it is 16bit code and TB have failed to provide a 32bit equivalent meaning that it can't be used by 32bit programs. Thus the AW2WF16 program, which is acting as a middle layer between AWAVE.EXE and WFGATE.DLL.

e) Advanced Gravis GF1 synths:

Q   I can only hear the first layer in a multi-layered instrument!

A   The GUS do not support multi-layering.

Q   Some regions are 'mute' in large instruments!

A   The current GUS drivers seems to be limited to max. 16 regions per instrument. Even though the .PAT file format supports more, only the first 16 regions will be used (and heard, the rest will be mute).

f) Advanced Gravis InterWave synths:

Q   Auditioning doesn't work! It plays the wrong instruments!

A   Make sure that you have the version 2.0 or later soundcard drivers.

Q   I can't audition this really large instrument, why?

A   The amount of memory on the soundcard can currently not be detected so the program never knows if an instrument will fit in memory or not. It will always try to load instruments up to 4MB in size (with 16bit samples always used during auditioning) and never any instruments larger than that

Q   It complains that it can't find 'IWINIT.EXE'. What can I do?

A   This file must be placed in the same directory as your IW.INI file. On most systems it already put there by the UltraSound installation program is there but if it not, then locate it and place a copy of it there.

Q   I have turned on auditioning and selected the GUS PnP as auditioning device, but now *Awave* just 'freezes up' whenever is select anything the tree-view list! Help!

A   If you look in the task bar (or at the minimized icons if you're a Win3.1 user) you'll see that a DOS-box having finished running IWINIT.EXE. Close down the DOS box manually and *Awave* should 'unfreeze'. To avoid this situation in the future, locate IWINIT.EXE on your disk using the Windows Explorer. Click the right mouse button on it and select properties. Select the 'Program' tab. Check the 'Close on exit' box. Click 'OK'. This assumed you were using Windows 95 or NT 4.x; if you're a Windows 3.1 or NT 3.x user you would use the PIF editor to do something similar (consult your Windows manual for help about the PIF editor).

Q   Why does it have to use IWINIT.EXE?

A   This is the only way that I currently have found to force the drivers to reread certain information that is crucial to making auditioning work...

F.A.Q. - Windows 3.1 issues

Q Program complains about not being able to load COMCTL32.DLL when I try to start it.
A You must have a Win32s version that supports the common controls introduced in Windows 95.When this was written, this means Win32s v1.3. Look in the WIN32S.INI file in the ...\ WINDOWS\SYSTEM directory to determine what version of Win32s you have. If you need an update you can get it from Microsoft, or, you can download it from my WWW-page.

Q The Midi SDS transfer and/or the auditioning stuff doesn't work.
A No. For some strange reason 32bit multimedia callbacks doesn't work on many Win32s systems, Sorry...

Q This automatic file association stuff doesn't seem to work.
A No. It is disabled under Win32s because it was designed for Windows 95 and under 3.1 it really clutters up things and doesn't work that well; so you'd probably not want it after all. You can manually associate files in the File Manager if you want to.

Q It just hangs when I try to start *Awave*!
A It seems like Win32s isn't compatible with a few systems (say 1%). If this is the case for you, then no other 32bit Win32s programs are working either.

Q Why does X happen when I try to do Y?
A Don't know, the Win32s library seems to be quite buggy on some systems (I guess it isn't a very high priority at Microsoft :-)).

Q Now I don't dare to install Win32s on my system!
A Take it easy. When you only run Windows 3.1 programs, then Win32s will lie dormant and not affect your system whatsoever. It is first when you start a 32bit program that Win32s gets loaded into memory.

MISC:

FMJ-Software*!*

If you enjoy this program, please
support my efforts by registering it.

OF:

Press to get help on file formats...

Displays what file format type the currently selected file has been determined to be.

The % value indicates how certain the file format type detection was with 100% being very certain, and 0% being quite unsure...

Select the subformat or data format to use when saving waveforms.

Check to allow saving in mono format. If both the mono and the stereo boxes are checked then the most suitable alternative will automatically be chosen for every waveform in the file.

Check to allow saving in stereo format. If both the mono and the stereo boxes are checked then the most suitable alternative will automatically be chosen for every waveform in the file.

PGP:

Shows if the program is registered or unregistered.

Click button to locate your personal registration keycode and remove all imposed limitations of the unregistered version of *Awave*.

Click button to uninstall this program.

This will clean up the registry entries for the program.
You can also use the Windows 95/NT control panel's install/uninstall feature to do exactly the same thing.

If you have a www-browser installed on your computer, then clicking here will get you to FMJ-Software's www site. There may find the latest news and updates to this program among other things.

Check this box if you want *Awave* to associate all its supported file formats with itself so that you may easily access *Awave* through '*play audio*' and '*convert*' commands on the Windows Explorer's right mouse button menu.
Note! If any other program is already associated with a file type, then that association will *not* be overwritten.
Note! if you ever uninstall *Awave* then these associations will be removed automatically.

Check to enable the main window toolbar.

PAP:

Check to enable auditioning.

Auditioning works by routing midi messages from a keyboard or other midi-in device to a wavetable synth, combined with automatic creation and loading of instruments and samples to the same wavetable synth.

Select the wavetable synth you want to use for the auditioning feature.

Select the midi input device you want to use for the auditioning feature.
Data from this device will be routed to your selected wavetable synth, and automatically translated to use the correct bank, program and channel.

Check to enable the virtual keyboard.

When enabled you can play on your computer keyboard instead of (or as a complement to) an external midi keyboard.

PPP:

Select the primary digital audio playback device you wish to use with ***Awave*** AudioPlayer. If this device is used when the AudioPlayer is launched then any other free device(s) will be used.

The AudioPlayer has an built in oscilloscope graph.

Selects the refresh rate of the AudioPlayer window's oscilloscope graph. Higher numbers gives a smoother look but taxes the CPU more.
If your computer is up to the task, set it to the monitor refresh rate (usually about 70Hz) or else set it to half or even a fourth of it.

Use Microsoft DirectSound services for waveform playback.
Selecting this will let you have multiple simultaneous AudioPlayers and the Awave software synth will me polyphonic rather than monophonic. It will also reduce latency (the time it takes before a sound starts playing) as well as much reduce the memory usage.
This option is only available if you have installed Microsoft DirectX v2 on your computer.

Use the old Windows multimedia services for waveform playback using an audio buffer queue.

Use the old Windows multimedia services for waveform playback without using a audio buffer queue. Select this only if you have trouble getting any sound from the AudioPlayer otherwise. Looping support in the AudioPlayer synth will also be disabled when this box is checked.

PHP:

You can use an external helper application for editing waveforms. Enter the file name and path for such an app here or click the [..] button to browse the disk for it. Then add any command line options necessary. Awave will launch the application with a .WAV file name appended last, e.g. "c:\windows\ sndrec32.exe –ASwitch New.WAV". The .WAV file name is generated automatically by Awave.

You can use an external mixer helper application. Enter the file name and path for such an app here or click the [..] button to browse the disk for it. It can then be launched by selecting 'Run mixer app' from the Options menu.

You can use an external on-screen keyboard helper application. Enter the file name and path for such an app here or click the [..] button to browse the disk for it. It can then be launched by selecting 'Run keybaord app' from the Options menu.

XGP:

Place any text comment here.

CGP:

Source file (import type) description.

Source file name and path.

Info on the number of top level items in the collection and their total size.

CPP:

A list of what instruments have been assigned to what midi bank and program numbers.
You can change the program number for an instrument in the instruments property sheet's general tab.

Click to add a new layer to the instrument.

Click to delete the selected layer.

IGP:

Info on the number of layers in the instrument and their total size.

Select melodic instrument type.

Select drum kit instrument type. This also selects that it should reside in the drumkit rather than in the melodic bank and program space.

The Midi 'Bank' number for the instrument. Bank 0 is generally the standard General Midi bank. Usually Midi controller CC00 controls the low order 7 bits of the bank number and contoller CC32 controls the high order 7 bits, i.e. Awave_bank = CC32_bank*128 + CC00_bank.

The Midi Program Change number for the instrument. Note that programs are numbered from 0 to 127 (some programs and synthesizers number them from 1 to 128 instead).

Controls to what degree the signal is sent to the reverb effects processor (if one is present).

Controls to what degree the signal is sent to the chorus effects processor (if one is present).

ILP:

List of layers used in instrument. Doubleclick to edit the layer keymap.

Press to add a new layer.

Press to delete the selected layer.

Press to add the currently selected layer in the pick list to the instrument.

The pick list: select the layer that will be added if you press the button to the left.

LKP:

The note name of the key currently under the cursor.

Click to create a new range object for the selected key range.

Click to clear (remove any range assignemts for) the keyrange selected.

Click to leave key range unmodified.

Select the low key of a keyrange.

Select the high key of a keyrange.

WGP:

Enter the sampling rate in Hz (number of recorded wavesamples per second). Valid range is 100.100 000 Hz.

Dome info about the waveform data format.
SourceBitsPerSample : ExpandedToBitsPerSample, NoOfWavesamples, Signformat, Channelformat

Root key.

This is the midi-key number at which the waveform is played back at its original sample rate.

Fine-tune.

Adjust the Root Key in steps of 1/1200th of an octave.
Range is -99..99 cents.

Enter relative volume damping.

Range is -96.0..0.0 dB (virtual silence..max volume) .

WEP:

Controls the position (offset) of the part of the waveform that is displayed.

Displays the waveform. Red is True RMS (Root of Mean of Squares) display, Light blue dots is maximums (P mode). Green is absolute current (I mode). Blue bars are loop points.

Controls the zoom in state of the waveform window.

Selects the power vs. time graph.

Selects the current vs. time graph.

Play the waveform using the *Awave* Audio Player.

Bring up the audio processing dialog.

Cut away the selection and place on the clipboard.

Copy the selection to the clipboard.

Paste waveform from the clipboard. It will be pasted 'over' the current selection.

Insert waveform from the clipboard. The insertion point will be the start of the current selection.

Select the whole waveform.

Displays the length in seconds or in millisecons of the current selection, or if nothing is selected, of the whole waveform.

WLP:

How the waveform looks like when its looping! They 'dark' parts are just before the waveform first enters resp. leaves the looping portion. The 'light' part is the actual looping section with the loop point break in the middle (this should be much easier to understand by looking at the graph and trying moving the loop point sliders than trying to explain it in words…)

Loop start point.

Also called repeat from point. The loop start point is the first wavesample *in* the loop range.
Note that fractional loop points is supported. This is used by synthesizers that can interpolate between samples.
The point is expressed in number of wavesamples from the beginning of the waveform, starting at zero.

Loop end point.

Also called repeat point. The loop end point is the first wavesample *after* the loop range.
Note that fractional loop points is supported. This is used by synthesizers that can interpolate between samples.
The point is expressed in number of wavesamples from the beginning of the waveform, starting at zero. It may not be less than the loop start point.

Fine-tunes the position of the loop start point.

Fine-tunes the position of the loop end point.

Zooms the two loop split displays in the time domain.

Zooms the two loop split displays in the amplitude domain.

Check to enable sampled release.

This means that the waveform loops while the key is pressed down, but stops looping and plays to the end of the sample (remember there may be sample data past the loop end point) when it is released. When this is not checked, it loops until it the amplitude envelope has faded it out upon key release.

Check to enable reverse looping.

This makes the loop go backwards instead of forwards.
This option may not be available on some sound hardware (including the *Awave* software synth)..

Check to enable looping.

When looping is on, the waveform is looped between the loop start and loop end points, as long as the midi key is pressed.

Check to enable bi-directional looping.

Bi-directional looping means the loop direction alternates between forwards and backwards. This option may not be available on some sound hardware (including the *Awave* software synth)..

Press to start/stop playing loop.

WSP:

Select the wavesample data type.

Check if signed wavesamples.

Check to perform a byte swap when reading 16bit data (necessary when reading 'Motorola byte order' data).

Check if stereo wavesamples.

Source file name, and the positions and size of the waveform data.

Enter the size of the file header (the number of Bytes preceding the actual sample data)..

Enter waveform size (number of Bytes of sample data).

The value is automatically truncated if it extends beyond the file length.
If all data, after the (optional) file header, is sample data, then you need not change this.
Note that 16bit data is 2 Bytes/Sample/Channel.

AGP:

Click to enable articulation.

Panning is the position of the waveform in the stereo field.
Range is -50.0% for full left, through 0.0% for centered, to +50.0% for full right.

Describes how the key on velocity affects the volume of the waveform - I think (it is a 'DLS level 1' parameter that I am not too sure about...)

Enter Volume LFO frequency.

Enter Volume LFO start delay.

Enter the scaling (amplitude) of the Volume LFO.

Enter how much the Midi ModWheel should affect the Volume LFO amplitude.

Enter Pitch LFO frequency.

Enter Pitch LFO start delay.

Enter the scaling (amplitude) of the Pitch LFO.

Enter how much the Midi ModWheel should affect the Pitch LFO amplitude.

The second envelope generator modifies the pitch.

The first envelope generator modifies the volume.

Select the range for the pitch envelope (controlled by EG 2). The maximum or minimum values of the envelope will be 0 and this pitch range. Setting a positive value will thus have the EG2 increase the pitch wile a negative will have it decrease the pitch. It is thus not posible to have a pitch envelope with both positive and negative values.

AEP:

What this envelope generator connects to (what it modifies).

A graph displaying the selected envelope generator curve.

Zoom the envelope graph in the time domain.

Set the value for the currently selected edit item (attack, or decay, or …).

Select to edit the attack time.

Select to edit the decay rate.

Select to edit the sustain level.

Select to edit the release rate.

Select to edit how much the midi key-down velocity affects the attack time (the faster the key was pressed, the faster attack rate).

Select to edit how much the midi key number, affect the decay time (the higher key number, the faster decay rate).

RGP:

Select which waveform you want to assign to this region.

Check to override certain waveform parameters.

Select the lowest (inclusive) midi key number in the range of this region.

Select the highest (inclusive) midi key number in the range of this region.

Select the lowest (inclusive) midi velocity in the range of this region.

Select the highest (inclusive) midi velocity in the range of this region.

Select the key group this region should belong to, or none if no key grouping is desired.

Click to see the properties for the linked waveform.

MGP:

Info about the midi sequence.

PLAY:

Name of the waveform.

Info about the waveform.

Press to start or stop playing the waveform.

Go to the next waveform in a sequence. Closes the Audio Player if this is the last waveform.

An oscilloscope!

The current playing position.

The length in seconds of the waveform.

Select to pause the oscilloscope graph.

Select to set waveform display mode.

Select to set spectrum analyzer mode.

File Formats

When opening files, *Awave* tries to determine the file type, not only by looking at the filename extension, but also by looking for 'signatures' inside the file. Thus, even if it has an incorrect extension, it can, for file types marked with ꜱᴜᴛᴏ, be correctly identified. For formats that are not possible to detect in this way you will have to pay attention to using correct file extensions. If a file is not of a recognized format, it is treated as raw audio data and the waveform properties source tab pops up querying you for additional information about the data format.

ave recognizes the following import file formats:

669	669 tracker module. ꜱᴜᴛᴏ	
AIFC	Compressed Audio Interchange File Format.	
AIFF	Audio Interchange File Format.	
AIS	Velvet Studio Instruments.	
ALAW	European telephony format audio.	
AMS	Extreme's Tracker module format.	
AMS	Velvet Studio Module.	
ASE	Velvet Studio Sample.	
AU	Sun/NeXT/DEC Audio file.	
AVI	MS Audio Video Interleave file.	
AVR	Audio Visual Research sound file.	
C01	Typhoon wave files.	
CDR	Raw Audio-CD data.	
DCM	DCM module format.	
DEWF	Macintosh SoundCap/SoundEdit recorded instrument format	
DIG	Digilink format.	
DIG	Sound Designer I audio file.	
DLS	DownLoadable Sounds.	
DMF	Delusion/XTracker Digital Music Fileformat.	
DSF	Delusion/XTracker Digital Sample Fileformat.	
DSM	Digital Sound Module tracker format.	
DTM	DigiTrekker module.	
DWD	DiamondWare Digitized file.	
EDA	Ensoniq ASR disk image.	
EDE	Ensoniq EPS disk image.	
EDK	Ensoniq KT disk image.	
EDQ	Ensoniq SQ1/SQ2/KS32 disk image.	

EDS	Ensoniq SQ80 disk image.
EDV	Ensoniq VFX-SD disk image.
EFA	Ensoniq ASR file.
EFE	Ensoniq EPS file.
EFK	Ensoniq KT file.
EFQ	Ensoniq SQ1/SQ2/KS32 file.
EFS	Ensoniq SQ80 file.
EFV	Ensoniq VFX-SD file.
EMD	ABT Extended MoDule.
ESPS	ESPS audio files.
EUI	Ensoniq EPS family compacted disk image.
F2R	Farandoyle linear module format.
F3R	Farandoyle blocked linear module format.
FAR	Farandoyle tracker module.
FFF	GUS PnP bank file format.
FNK	FunkTracker module format.
FSM	Farandoyle Sample format.
FZB	Casio FZ-1 Bank dump.
FZF	Casio FZ-1 Full dump.
FZV	Casio FZ-1 Voice dump.
G721	Raw CCITT G.721 4bit ADPCM format data.
G723	Raw CCITT G.723 3 or 5bit ADPCM format data.
GKH	Ensoniq EPS family disk image file.
GSM	Raw GSM 6.10 audio stream.
GSM	US Robotics voice modems / QuickLink .GSM format.
HCOM	Sound Tools HCOM format.
IFF	Interchange file format.
INI	MWave DSP synth's mwsynth.ini GM-setup.
INI	Gravis UltraSound bank setup.
INRS	INRS-Telecommunications audio.
INS	Ensoniq EPS family instrument.
INS	Sample Cell / II Mac instrument.
INS	Sample Cell / II PC instrument.
IST	Digitrakker Instrument File.
IT	Impulse Tracker module/instrument/sample.
K25	Kurzweil 2500 sample.

KR1	Kurzweil 2000 sample (multi-floppy).
KRZ	Kurzweil 2000 sample.
MAT	Matlab variables binary file.
MAUD	MAUD sample format.
MDL	Digitrakker Module.
MED	OctaMED tracker module.
MOD	Amiga and PC tracker module.
MOV	Apple QuickTime audio.
MP1	MPEG audio stream, layer I
MP2	MPEG audio stream, layer II
MP3	MPEG audio stream, layer III
MPA	MPEG audio stream, layer I, II or III
MTM	MultiTracker Module.
MUS10	Mus10 audio.
NIST	NIST Sphere audio.
O01	Typhoon vOice file.
OKT	Oktalyzer tracker module.
PAC	SB Studio II package.
PAT	Advanced Gravis Ultrasound / Forte tech. Patch.
PBF	Turtle Beach Pinnacle Bank File.
PCM	OKI MSM6376 synth chip PCM format.
PLM	DisorderTracker2 module.
PLS	DisorderTracker2 sample.
PPF	Turtle Beach Pinnacle Program File.
PRG	WAVmaker program.
PSB	Pinnacle Sound Bank.
PSION	PSION a-law audio.
PSM	Protracker Studio Module Format.
PTM	Poly Tracker module.
RAW	Raw signed PCM data.
SAM	Signed 8bit Sample data.
SB	Raw Signed Byte (8bit) data.
SBK	Emu SoundFont v1.x Bank files / Creative Labs SB AWE 32.
SD	Sound Designer I audio.
SD2	Sound Designer II flattened file.
SD2	Sound Designer II data fork.
SDK	Roland S-series floppy disk image.

SDS Raw Midi Sample Dump Standard file.

SDW Raw Signed DWord (32bit) data.

SDX Midi Sample Dump Standard files as compacted by SDX.

SF IRCAM SoundFile format.

SF2 Emu SoundFont v2.0 file.

SFD SoundStage Sound File Data.

SFI SoundStage Sound File Info.

SFR Sonic Foundry Sample Resource.

SMP Samplevision format.

SMP Ad Lib Gold Sample.

SND Raw unsigned PCM data.

SND AKAI MPC3000 sample.

SNDR Sounder sound file.

SNDT Sndtool sound file.

SOU SB Studio II sound.

SPD Speach Data file.

SPL Digitrakker Sample.

SPPACK SPPack sound sample.

STM Scream tracker v2 module.

S3I Scream tracker v3 instrument.

S3M Scream tracker v3 module.

SVX Interchange file format, 8SVX/16SV.

SW Raw Signed Word (16bit) data.

SYW Yamaha SY-series wave files (really named W??).

TXT Ascii Text formatted audio data.

TXW Yamaha TX16W wave files (really named .W??).

UB Raw Unsigned Byte (8bit) data.

UDW Raw Unsigned DWord (32bit) data.

ULAW US telephony format (CCITT G.711) audio.

ULT UltraTracker modules.

UNI MikMod 'UniMod' format.

UW Raw Unsigned Word (16bit) data.

UWF UltraTracker Wave File.

V8 Covox 8bit audio.

VAP Annotated speech.

VOC Creative labs sound.

VOX Dialogic adpcm.

VOX Talking Technology Incorporated file.

W??	Yamaha waveforms (see TXW, SYW).
WAV	Ms Windows PCM wave.
WFB	Turtle Beach WaveFront Bank (Maui/Rio/Monterey).
WFD	Turtle Beach WaveFront Drum set (Maui/Rio/Monterey).
WFP	Turtle Beach WaveFront Program (Maui/Rio/Monterey).
XM	Fast Tracker 2 extended module.
XI	Fast Tracker 2 instrument.
*	Unrecognized formats prompts user to supply basic info. (

File Formats

When saving files, select the desired File Type in the appropriate list in the Save As dialog, or just type in a file name with an extension matching one of the supported export types. Note that most of the formats can only handle a subset of the item types as denoted by the symbols below.

ave _can export to the following formats:_

AIF   Audio Interchange File Format, .

ALAW   Raw A-law data,
.

AU   Sun Audio format, linear, μ-law or A-law,
.

DLS   DownLoadable Sounds,
,
,
.

DWD   DiamondWare Digitized audio,
.

EFE   Ensoniq EPS family instrument,
,
.

FFF   GUS PnP bank file format,
,
.

FZB   Casio FZ-1 Bank dump,
.

FZV   Casio FZ-1 Voice dump,
.

G721   Raw CCITT G.721 4bit ADPCM format data,
.

G723   Raw CCITT G.723 3 or 5bit ADPCM format data,
.

GSM   Raw GSM 6.10 audio stream,
.

GSM   US Robotics voice modems / QuickLink .GSM format,
.

IFF   Interchange file format,
.

INI   Gravis UltraSound bank setup extract plus patch files,
.

INS   Sample Cell / II Mac instrument,
.

INS   Sample Cell / II PC instrument,
.

MAT   Matlab variables binary file,
,
.

MWS MWave DSP synth's instrument extract,
✖,
✖.
PAT Advanced Gravis UltraSound / Forte tech. patch file,
✖,
✖.
PBF Turtle Beach Pinnacle Bank File,
✖,
✖.
PRG WAVmaker program file,
✖.
S3I Scream tracker v3 instrument,
✖.
SB Raw Signed PCM Byte (8bit) data,
✖.
SBK Emu SoundFont v1.0,
✖,
✖,
✖.
SD Sound Designer I audio file,
✖.
SD2 Sound Designer II flattened file,
✖.
SD2 Sound Designer II data fork,
✖.
SDS Midi Sample dump standard,
✖.
SF2 Emu SoundFont v2.0 / AWE32 file,
✖,
✖,
✖.
SND AKAI MPC3000 sample,
✖.
SW Raw Signed PCM Word (16bit) data,
✖.
TXT Ascii Text parameter description,
✖,
✖,
✖.
TXT Ascii Text formatted audio data,
✖.
UB Raw Unsigned PCM Byte (8bit) data,
✖.
ULAW Raw μ-law (CCITT G.711) data,
✖.
UW Raw Unsigned PCM Word (16bit) data,
✖.
VOC Creative labs sound file,
✖.
VOX Dialogic adpcm file,
✖.

W01   Yamaha TX16W wave,
✖.

W01   Yamaha SY-series wave,
✖.

WAV   Ms Windows PCM wave file, linear, μ-law, A-law or GSM,
✖.

WFB   Turtle Beach WaveFront Bank,
✖.

WFD   Turtle Beach WaveFront Drumkit,
✖.

WFP   Turtle Beach WaveFront Program,
✖,
✖.

XI   Fast Tracker 2 instrument,
✖,
✖.

✖   *indicates that the format supports saving waveforms.*

✖   *indicates that the format supports saving instruments.*

✖   *indicates that the format supports saving collections.*

✖

*: *Unrecognized file types*

If a file cant be determined to be one of the, by this program, supported file types, then the waveform properties box will appear. Select the source tab and supply information on the files data type.
Note! if you get it wrong, you can always use the waveform properties dialog again and try some other settings.

Features:

8 / 16 bits / µ-Law / A-Law, mono / stereo.

.669: 669 tracker module (PC)

Features:

8 bits, mono, loop, name, bank.

.AIF/.AIFF/.AIFC/.C01: Audio Interchange File Format (Amiga, Atari, SGI,Mac)

AIFC is an extension of the original AIFF format also supporting compressed sample data.
The .C01 files are used by the Typhoon OS for the Yamaha TX16W.

Features:

1..32bits / IMA / DWVW compression / μ-law encoding, loop, keyrange, mono / stereo, name.

Unsupported features:

Apple MACE and ACE compressions can not be supported for patent reasons.

.ALAW: European Telephone format (CCITT G.711) audio

Features:

A-Law, mono, 8kHz.

.AMS: Extreme's Tracker format (PC)

Features:

mono, 8,16 bits, name, bank.

Unsupported features:

Compression.

.AMS/.AIS/.ASE: *Velvet Studio Module/Instrument/Sample (PC)*

Features:

     8,16 bits, mono, name, loop, instrument, collection.

.AU: Audio file (Sun/NeXT/DEC)

Features:

8,16,24,32 bits / μ-law / A-law / CCITT G.721 (4bits) & G.723 (3,5bits) ADPCM / IEEE FP (32, 64bits) FP encodings and compressions, mono / stereo.

Unsupported features:

> 2 channel files.
Other compression schemes.

.AVI: MS Audio Video Interleave file (PC)

Features:

Same as for .WAV files

Unsupported features:

Of course, only the digital audio streams are read, not video or such...

.AVR: Audio Visual Research sound file (Macintosh)

Features:

    8,16bit, mono / stereo, name, loop.

.CDR: Raw Audio-CD format data file

This is raw CD-*format* audio, and not raw CD-*data* which also contains a lot of error correction and formatting information, i.e. you can not read directly from an audio CD but if you have made a raw sound data dump with another program, you can give it an .CDR extension and Awave will read it properly.

Features:

16 bits, stereo.

.DCM: DCM module format

Features:

8,16bits, mono, loop, collection.

.DEWF: SoundCap / SoundEdit recorded instrument format (Macintosh)

Features:

8bit, mono, loop.

.DIG: ***Digilink format (Broadcasting)***

Features:

16bit, stereo.

.DLS: DownLoadable Sounds level 1 (PC)

Awave reads and write files conforming to a *beta version* of the forthcoming DLS v1.0 file format specification. An update to *Awave* will be released when the final specification is available.

Saving to the 'DLS level 1' file format will give you files compatible with the DLS synthesizer architecture level 1, which is designed to be compatible across a wide range of synthesizers.

You can also save what *Awave* calls ' *DLS extended level 1*' files. These conform with the general v1.0 DLS file format but ignore the restrictions of the architecture level 1 (and thus may, or may not, be compatible with future DLS atchitecture level 2). These extended files can store all data representable in *Awave*, except i) no fractional loop points, ii) no advanced loop types (other than no loop, or forward loop), and iii) frequency and start delays are shared between the pitch and volume lfo's.

When using pure DLS level 1 keep in mind the following:
 - Only one layer per instrument can be used.
 - A maximum of 16 regions can be used in a melodic instrument, 128 in a drum set.
 - Only instrument global articulation is allowed for melodic instruments.
 - A regions velocity range is always 0..127.
 - Panning is ignored for melodic instruments.
 - Only mono waveforms can be used.
 - Only unlooped and forward looped waveforms are supported.
 - The minimum required RAM memory of a DLS-1 synth is 256K *Samples*, if you use larger collections, data may be discarded on synths with little memory.
 - The volume and the pitch lfo's will use the same frequency and start delay values (a weighted average will be used if they are not equal for the two lfo's).

Features:

8,16 bits, mono, loop, envelopes, lfo's, collection, drumkits, instruments, layers, ...

.DMF/.DSF: Delusion/XTracker Digital Music/Sample Fileformat (PC)

Features:

8,16bits, mono, loop, name, instruments, collection.

Unsupported features:

Packed samples.

.DSM: Digital Sound Module tracker format

Features:

8 bits, mono, name, loop, collection.

.DTM: DigiTrekker module

Features:

8,16bits, mono, loop, name, collection.

.DWD: DiamondWare Digitized file

Features:

8/16bits, mono/stereo.

.EDx,GKH,EUI: Ensoniq disk image (MIDI)

The Ensoniq files are usually encountered in form of .GKH disk image files (with an additional header) as well as in the slightly compacted .EUI disk image format. You may also find the .EDx files, x = A,E,K,Q,S,T,V.

Awave can only read contiguous subfiles in Ensoniq disk images. If *Awave* can't handle a certain file (because it's fragmented on the disk), you can extract the individual instrument files to .INS or .EFE files with a utility called EPSDISK (not included with *Awave*). INS files are the same as an EFE files except the latter may also contain EPS sequenced songs, bank info or such, instead of an instrument.

Features:

16 bits, mono, loop, names, fine-tune, layers, instruments, collection.

Unsupported features:

Envelopes, .EDT (Ensoniq TS) files.

.EFx,.INS: Ensoniq instrument file (MIDI)

INS files are the same as an EFx files except the latter may also contain EPS sequenced songs, bank info or such, instead of an instrument.

Note! If an Ensoniq instrumen uses a 'custom pitch table' or is 'unpitched' instead of using the 'standard pitch table', then it will be emulated by trying to set the region root key ovverrides values. This 'emulation' will only correct for the 'middle key' in the region but it will usually sound ok over the whole region and drumkits (that often are 'unpitched') usually have regions that only span one key anyway.

Features:

16 bits, mono, loop, names, fine-tune, layers, instruments.

Unsupported features:

Envelopes, .EFT (Ensoniq TS) files.

.EMD: ABT Extended MoDule format

Features:

16bit, mono, loop, name, collection.

.ESPS: ESPS audio file

Features:

8,16,32bit integer / 32,64bit FP, mono.

.FAR,.FSM,.F2R,.F3R: Farandoyle tracker format (PC)

Features:

8,16 bits, mono, loop, name, collection.

.FFF+.DAT: GUS PnP bank file (PC)

Note! The .FFF file contains all 'instrument parameters' and stuff while the accompanying .DAT file contains the actual sample data which is to be downloaded to the synth's memory.

Features:

8/16 bit, mono, loop, lfo, amplitude envelope, layers, instruments, drum kits, collection.

Unsupported features:

Pitch envelope.

.FNK: FunkTracker module format (PC)

Features:

8 bit, mono, loop, name, panning, collection.

.FZB/.FZF/.FZV: Casio FZ-1 Bank / Full / Voice FZDUMP/FZCom dump

These files are the result of dumping one or more Casio FZ-1 'voices' (= waveforms) and/or 'banks' (= instruments) to a PC using either a midi cable or a parallel cable connection and a program called FZDUMP or one called FZCom.

Features:

16 bit, mono, loop, name, instruments.

Unsupported features:

Envelopes and lfo's.

.G721:   Raw CCITT G.721 4bit ADPCM format data

Note! The first sample is assumed to be in the high-nibble, the second in the low-nibble.

Features:

ADPCM 4bit lossy compression, mono.

.G723: Raw CCITT G.723 3 or 5bit ADPCM format data

Note! When you read one of these files a dialog pops up where you manually must chose if it is 3 or 5bit per sample data, this can't be autodetected…

Features:

ADPCM 3 or 5bit lossy compression, mono.

.GSM: GSM 6.10 audio stream

Note! These files can't be autodetected so be sure to use the .GSM extension. The data is assumed to be in 33Bytes/frame packets where the first nibble (4bits) is ignored.

Features:

GSM compression, mono.

.GSM: US Robotics voice modems / QuickLink .GSM format

Features:

GSM compression, mono.

.HCOM: Sound Tools HCOM format (Macintosh)

Features:

Huffman compression, mono.

.IFF/.SVX: Interchange File Format: 8SVX or 16SV data (Amiga)

The 16SV flavor is rather uncommon and is, I think, only used by some PC trackers. Thus if you save 16 bit waveforms as .IFF files, don't expect them to be readable by most .IFF readers.

Features:

8,16 bits, loop, name, mono / stereo.

.INI: *Gravis UltraSnd.Ini bank setup (PC)*

You can read the whole melodic bank 0 setup with all the associated patches from the UltraSnd.Ini file. You can also write 'extract' .INI files for a complete bank that you can then paste into your UltraSnd.Ini file, this will also save all the patches referenced in the bank to .PAT files (they will be automatically named based on the name of the .ini extract file).

Features:

Same as for .PAT + collection + drumkits.

.INRS: INRS-Telecommunications audio file

Features:

16 bits mono.

.INS:Sample Cell / II PC or Mac instrument

There are two different (incompatible but very similar) varieties of these files, a PC format and a Mac format. These files contains one instrument definition (per file) and references a bunch of external waveform files. For the PC format, the external files is .WAV files. For the Mac format they can be either .AIF(F) or .SD2(F) files. When reading an .INS file you should put all these external files into the same directory as the .INS file. It is very important that these external files retain the original filename that they had when the .INS file was written or Awave will not be able to find them. Thus when transferring Mac files to a PC, take xtra care to retain the 'long' file names. In case of .SD2 file you must transfer it to a 'flattened' MacBinary format file.

Features:

8/16bits mono, instrument.

Unsupported features:

lfo's and envelopes.

.IT: *Impulse Tracker module/instrument/sample (PC)*

Features:

8,16 bits, mono, stereo, loop, name, instrument, collection.

Unsupported features:

MMCMP compressed files.

.KRZ/.KR1/.K25:   *Kurzweil K2000/K2500 instrument file (MIDI)*

The K2000 file format is considered proprietary by Kurzweil; so any resemblance with the .KRZ file format this program reads is purely coincidential :-/. If one, against better knowing should try to use this program to read the former file type, one shall find that they match fairly closely :-D, of course, sometimes things may go a bit wrong here and there...

To read multi-floppy files. Rename the file parts into .KR1 for the first floppy, .KR2 for the second et c. Place all the parts (.KR1, .KR2, et c.) in one and the same directory on your hard disk. Then open the .KR1 file to read them all.

.K25 files, intended for the K2500, seems to be compatible with the K2000's .KRZ files.

Features:

16 bits, mono, loop, names, fine-tune, layers, instruments, collection.

Unsupported features:

Envelopes, stereo… Drumkits may sound out of tune.

.MAT: *Matlab variables binary file*

Note! 8bit is saved as unsigned 0..255 and 16bit as signed -32768..32767, however Matlab may think it's too loud when using sound(var,sr) so you may need to divide the sound data vector it by 8 or something.
Note! When reading floats they are expected to be in the range [-1, 1]
Note! Stereo files are read/written as complex vectors.

Features:

8,16bits / floats, mono / stereo, collection.

.MAUD: MAUD sample format

Features:

8,16 bits / μ-Law / A-Law, mono / stereo, name.

.MDL/.IST/.SPL: *Digitrakker Module/Instrument/Sample File (PC)*

Features:

    8,16bits, mono, loop, name, huffman packing, collection.

.MED:　OctaMED tracker module:MMD0\MMD1\MMD2 (Amiga)

Note! The old MED tracker .MED format is not supported, if you have a description of this format, please send it to me.

Features:

8bit, loop, name, collection.

.MOD: Module file (Amiga, PC)

Recognizes the following flavors:
M.K., M!K!, 4CHN, 6CHN, 8CHN, 10CH, 12CH, 14CH, 16CH, 24CH, 32CH, FLT4, FLT8, OCTA, CD81, EX08.

Features:

8 bits, mono, loop, name, fine-tune, collection.

.MOV: Apple QuickTime audio (Macintosh)

Reads the audio track form QuickTime movies.

Features:

8/16 bits/μ-law, mono/stereo.

Unsupported features:

Compressed formats.

.MTM: Mtm tracker module (PC)

Features:

8, 16 bits, mono, loop, name, collection.

.MPA/.MP1/.MP2/.MP3:   MPEG audio stream, layer I, II or III

Note! These files can't be autodetected so be sure to use any of the above file extensions.
Note! If it is a large file, then you will need alot of memory to decode…

Features:

16 bits, mono / stereo.

.MUS10: Mus10 audio file (PDP-10)

Features:

12,16,18,20bit, mono/stereo.

MWSYNTH.INI: MWave DSP synthesizer's GM bank setup (PC)

The software synthesizer module for the MWave DSP uses this file to construct instruments from usual .WAV files.

Features:

8,16 bits, loop, name, instruments, layers, collection.

.MWS: MWave DSP synthesizer's MWSYNTH.INI instrument extract (PC)

The software synthesizer module for the MWave DSP uses the MWSYNTH.INI file to construct instruments from usual .WAV files. When exporting to the MWS format, **Awave** outputs a bunch of .WAV files as well as a .MWS text file containing an instrument extract that can be manually pasted into the MWSYNTH.INI file.

Features:

8,16 bits, loop, name, layers, instruments.

.NIST: NIST SPHERE file

Features:

16 bits / μ-Law, mono / stereo.

.O01: Typhoon vOice file (Yamaha TX16W)

Note! The actual waveform data are stored in accompanying .C01 file(s).

Features:

DWVW compression, mono, loop, instrument.

.OKT: *Oktalyzer tracker module (Amiga)*

Features:

8 bits, mono, loop, name, collection.

.PAT: Advanced Gravis UltraSound GF1 / Forte tech. patch file (PC)

Note! The current UltraSound drivers do not support multi layering and no max. 16 waveforms per instrument.

Features:

8,16 bits, mono, loop (incl. fractional), envelopes, lfo, names, instruments.

.PBF/.PPF: *Turtle Beach Pinnacle Bank/Program File (PC)*

The current version of this format stores waveforms in external .WAV files that must be placed in the same directory as the .PBF file. Future versions will support 'embedded' waveforms where they are stored inside the .PBF file. PPF files do not actually exist when this was written but I am told they will be the same as PBF only that they must only contain a single instrument…

Features:

8/16, name, loop, layers, instruments, drum-kit, collection.

Unsupported features:

Envelopes, lfos, effects.

.PCM: OKI MSM6376 synth chip PCM format

This format supports the following sample frequencies: 4000, 5300, 6400, 8000, 10600, 12800, 16000, 21200, 32000 Hz. When saving files, the nearest supported frequency will be used and the waveform first resampled if it is not exactly the same frequency. To explicitly save in a certain frequency, first *manually* resample the waveform to that frequency.

Features:

16 bits, mono.

.PLM/.PLS: DisorderTracker2 module/sample (PC)

Features:

8,16 bits, mono, loop, name, collection.

.PRG: WAVmaker program file (PC)

Note! when importing the referenced .WAV files must be in the same directory as the .PRG file and when exporting, the .WAV files will also be put in the same directory.
Note! If you want to use Awave to create .PRG instruments, then, for best results when setting loop points, make sure that you _first_ resample the waveform(s) (by waveform->Properties->Wavform->Process->Resample...) into one of the rates supported by WAVmaker: 11025, 22050 and 44100 Hz. If you do not, then they will automatically be resampled when you save the .PRG file..

Features:

8/16bits,mono/stereo,loop,name,instrument.

.PSB: Pinnacle Sound Bank (PC)

Note! This format is basically a list with assignments of .WAV files to midi program numbers. Make sure you have the referenced .WAV files in the same directory.

Features:

8/16bit, mono/stereo, loop, name, instrument, drumkit.

.PSM: Protracker Studio Module Format

Features:

8,16bit, mono, loop, name, collection.

.PSION: PSION alaw file

Features:

A-law, mono.

.PTM: Poly Tracker module

Features:

8,16bit, mono, loop, name, collection.

.SAM: Raw signed PCM sample file

Features:

8 bits, mono.

.SB, .SDW, .SND, .SW, RAW, .UB, .UDW, .UW: Raw PCM data format file

The file extension has the following meaning:
S   Signed data
U   Unsigned data
B   Byte (8bit) data
W   Word (16bit) data
DW   DWord (32bit) data
.SND is the same as :UB.
.RAW is the same as .SB.
Assumes 11khz mono, Intel byte order when reading. Use the waveform properties dialogs source tab to change any parameters that are not correct.

Features:

8,16,32 bits.

.SBK/.SF2:   EMU Sound Font Bank v1.x / v2.x file / Creative Labs SB AWE 32 (PC)

The .SBK format (Sound Font v1) has been replaced with .SF2 (Sound Font v2) which should be used for all new files. If you are used to the naming convetions of the Vienna 2.0 editor, then the following small 'dictionary' may be usefull:

Awave   *Vienna 2*
instrument   preset
layer   instrument
region   instrument zone
waveform   sample

Note that a layer in wave must contain regions with non-colliding keyranges which a Vienna instrument must not. Thus, when reading .SF2 file, *Awave* may split an instrument into several layers. Also, if there is any 'Vienna instrument' that is not used by any preset, then it won't be read.

Features:

16 bits, pitch, loop, name, lfo, envelopes, layers, instruments, drumkits, collection.

.SD, .DIG: *Sound Designer I audio file (Macintosh)*

Features:

 8,16 bits, mono, name.

.SD2: *Sound Designer II flattened/resource fork file (Macintosh)*

On the Macintosh these files are really two file 'forks'. When they are transferred to a PC these two forks may be flattened (preserving all data and parameters) or only the resource fork (containing the raw sound data) may be copied. if only the resource fork has been copied, then it is assumed to be 16bit mono data.

Features:

8,16 bits, mono/stereo.

They might go under another extension but SDK seems to be most common.
Note! The instrument key assignments may not always be read correctly so you may have to do some manual editing on that.
This is really two file formats, one 12bit used by Roland S50, S51, S550 and S330 synhtesizers and one 16 bit used by Roland S70, S750, S770 and S772 synthesizers.

Features:

12/16 bits, mono, loop name, fine-tune, instrument, collection.

Unsupported features:

A zillion different parameters.

.SDS, .SDX: *Midi Sample Dump Standard file (MIDI)*

SDS is a raw midi data dump of an SDS (Sample Dump Standard) transfer and can be up to twice as large as the actual waveform. SDX is a 'compacted' form that the program SDX uses to save SDS samples in order to avoid this problem. It also contains a sample name which SDS does not.

Features:

8..32 bits, mono, loop, (SDX only: name , note).

.SF: IRCAM SoundFile format

Features:

16 bits, name, note, mono / stereo.

Unsupported features:

Other bits/sample.
SFISFI   SoundStage Sound File Info (.SFI) + Data (.SFD) files.

.SFI, .SFD: SoundStage Sound File Info + Data file.

Note! These files must always come in a .SFI + .SFD pair! You should only load the .SFI file.

Features:

16 bits, name, note, mono / stereo.

Unsupported features:

Other bits/sample.
SoundStage Sound File Info (.SFI) + Data (.SFD) files.

.SFR: Sonic Foundry Sample Resource file.

Features:

8,16 bits, mono, stereo.

.SMP: Samplevision file (PC)

Features:

16 bits, mono, loop, name, note.

Unsupported features:

Only the first of the 8 possible loop points are considered.

.SMP: Ad Lib Gold Sample file (PC)

Features:

    8,16 bits, mono, stereo.

Unsupported features:

    Yamaha ADPCM compression.

.SND: AKAI MPC3000 sample file

The AKAI program (.PGM) files defining instruments (and referring to various .SND waves) are currently not supported. Any info on this format would be welcome.

Features:

16 bits, mono, loop, name.

.SNDR: Sounder sound file

Features:

8 bits, mono.

.SNDT: SndTools sound file

Features:

8 bits, mono.

.SOU,.PAC: SB Studio II sound/package file (PC)

Features:

8,16 bits, mono, loop, name, collection.

.SPD: Speach Data file (UNIX)

Features:

16 bits / mono.

Features:

8,16 bits / μ-Law / A-Law, mono.

.STM: ScreamTracker module file (PC)

Features:

8 bits, mono, loop, name, collection.

.S3M, .S3I: ScreamTracker v3 modules / Digiplayer sample (PC)

The .S3I files are usually named .SMP but they will be saved as .S3I in order to avoid confusion with Samplevision files..

Features:

8,16 bits, mono / stereo, loop, name, note, collection.

.SYW/.W??:   *Yamaha SY-series wave file (MIDI)*

The Yamaha 16-bit SY-series wave sample files are really named *.W??*. There's also *.T??* (All Data Files, often contains waves and are much more common than .W??; but aren't supported at the moment), *.J??* (Patch parameters?) and *.K??* (Sequenced songs). See also the 12bit format. *Note!* Before reading a directory of files on your synthesizer, make sure they have all unique extensions like .W01, .W02, .W02 et c.

Features:

16 bits, mono, loop.

.TXT: Ascii Text parameter description file

Contains human readable information about items at bank, instrument or waveform levels. Useful if *Awave* can't export directly to your destination format and you'll have to piece together instruments from, say, .WAV files.

Features:

Human readable information, no sample data.

.TXT: *Ascii Text formatted audio data*

Ascii text file with integer sample values in base 10. Samples are deliminated by either , (ascii no 44) or new-line (ascii no 10). Comments and lines not containing sample values must begin with a ; or a %.

For stereo files (with interleaved samples), you must manually set stereo in the waveforms properties box source tab.

Ex 1:

```
;signed 8bit, 3 samples
127   ; first sample
-3
15      ; last sample
```

Ex 2:

```
% This will be detected as unsigned 15bit, 5 samples
9,0,32767,1000,20001
```

Features:

1..32 bits, signed or unsigned (autodetect).

.TXW/.W??:   Yamaha TX16W wave file (MIDI)

The Yamaha TX16W (and probably also the other Yamaha 12bit samplers) waveform files are really named *.W??* (where ?? is a number). There's also *.F??* (Filter), *.V??* (Voice to midi no assign?),*.U??* (Performance settings) and *.S??* (System setup?) files which are of no use to you. See also the 16bit format.

Note! The format only supports 16, 33 or50kHz sample frequency and no tuning or rootkey parameters so when saving to TX16W .W01 files, the closest of the three rates will be selected and you must manually tune it when you're using it in an instrument on your synth... Alternatively you may resample it in *Awave* to either of 16,33,50kHz before saving it.

Note! Before reading a directory of files on your synthesizer, make sure they have all unique extensions like .W01, .W02, .W02 et c.

Note! When saving files, be aware that the TX16W only supports a maximum of 128K samples in the 'attack' plus 128K samples in the 'loop', if the waveform is longer then it will be truncated be Awave.

Features:

12 bits, mono, loop.

.ULAW: US Telephone format (CCITT G.711) audio

This is also the format used by the MIME mail encoding standards audio/basic attachments..

Features:

μ-Law, mono, 8kHz.

.ULT, .UWF: UltraTracker module / wave file (PC)

Features:

8,16 bits, mono, loop, name, collection.

.UNI: MikMod 'UniMod' tracker format (PC)

Features:

8,16bit, mono, name, loop, instruments, collection.

.V8: Covox 8bit file (PC)

Features:

8bit, mono.

.VAP: VAP annotated speech file

Features:

8bit, mono, name.

.VOC: Creative Labs sample file (PC)

Note! When exporting to the .VOC format, files will be written in a newer and much more capable subformat that may not be recognized by some old programs.

Features:

8,16 bits / μ-Law / A-Law, mono / stereo, name.

Unsupported features:

Other compression and packing schemes.

.VOX: Dialogic adpcm file

Features:

Adpcm compression (4bit), mono, 8kHz.

.VOX: Talking Technology Incorporated file

Features:

8bit, mono.

Unsupported features:

Compressed subformats.

.WAV: *Microsoft Wave file (PC, Windows, OS/2, NT)*

Supports the Sample (smpl) chunk (used by Turtle Beach Wave SE, WavePatch, Gravis PatchMaker and others), the Instrument (inst) chunk (used by IBM?) as well as 'cue '+'labl' chunks to support loop.
Also supports some info chunks (INAM, ICMT, ISFT).

Features:

1..32 bits PCM / μ-Law / A-Law, MS ADPCM (4bit) / Intel DVI/IMA ADPCM (2,3,4bit) / CCITT G.721 (4bit) & G.723 (3,5bit) ADPCM / Dialogic (4bit) ADPCM / GSM 6.10, MPEG audio encodings and compressions, mono / stereo, note, loop, rootkey, fine-tune.

Unsupported features:

Other compression schemes and formats.

.WFB, .WFP, .WFD: Turtle Beach WaveFront synthesizers / WavePatch file (PC)

WaveFront based synthesizers are currently the Maui, Tropez, Rio and Monterey.
.WFB files contains a whole Midi Bank, .WFP a single Program and .WFD a complete drum set.
Instruments using ROM samples, of course, can't be converted.
If the files doesn't use embedded waveforms, but external .WAV files instead, you must have
the .WAV files in either the exact path they where when the .WF? file was created, or, in the same
directory as the WF? file.
Currently *Awave* only saves files with embedded waveforms.

Note! TB Rio users should only save .WFP files as the .WFB exported by this program are 'Maui'
style and may not be loadable on the Rio.

Features:

8/16/μ-Law bits, name, note, loop (incl. fractional), envelopes, lfos, layers, instruments, drumkit,
collection.

Unsupported features:

Pre v1.2 files.

.XM, .XI: FastTracker 2 eXtende Module/Instrument (PC)

Features:

8,16 bits, mono, loop, name, instruments, panning, collection.

Unsupported features:

A bunch of ramping parameters (envelope, pan, vibrato...).