

## *Various wavetable instrument formats to*



## *Gravis Ultrasound Patch file format converter*

*Written by:*

*F. Markus Jönsson*

*Svedjevägen 7, 147 52 Tumba, SWEDEN*

*phone: +46-8-530 390 81*

*email: f93-maj@nada.kth.se*



### **Contents**

1. [Terms of Use](#)
2. [Step-by-Step Instructions](#)
3. [File Formats](#)
4. [Notes](#)
5. [Revision History](#)
6. [Adds](#)

*A big thanx should go to those people who have supported me with information and beta testing.*



## Terms of Use

x

### *Licensing agreement:*

Currently, this piece of software is being distributed as so called cardware, i.e. send me a postcard (by snail mail, i.e.) of your own choice, put down upon it a few lines of appreciation and I'll grant you the legal right to use this software as of this version.

Registered users of previous versions are hereby granted the right to use this version (without sending me another postcard), but note that future versions are not covered by above statement unless explicitly stated in the licensing agreement of those versions (i.e. in case my licensing policy should happen to change).

### *Disclaimer of Warranty:*

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.

### *Trademarks and stuff:*

All trademarks are the property of their respective owners.

### *Distribution:*

Please feel free to distribute this software at will. You may *not ever* charge any money for it though, not even to cover material, handling or transportation expenses of any kind.

The following files, and those only, must always be supplied:

2PAT.EXE	- Program executable.
2PAT.HLP	- Program documentation in WinHelp hypertext format.
FMJMTHRU.DLL	- FMJ Midi thru support (v1.1).
FMJULTRA.DLL	- FMJ Ultrasound.Ini manipulation support (v1.0).
BWCC.DLL	- Borland Windows Custom Controls (v2.0)
FILE_ID.DIZ	- BBS info file.

x

## Step-by-Step Instructions

✘

- 1) Run **2PAT**.
- 2) Select **Load instr.** and choose a file in the common dialog box.  
A scroll list with the following info for all instruments, contained within the file, will then be shown:

```
wind.panflute      1:12345  8  44
|                 | |      |  |
Instrument name  | |      |  Frequency in KHz
                   No of waves |      8/16 bits/sample
                               Instrument size
```

- 3) Select an instrument from the list.  
You can then play it if you have a midi-in device (of course, you must have an Ultrasound card as well).
- 4) Select **Save patch** or doubleclick on an instrument.  
A common dialog box will appear and prompt you for the file name to save to.  
Typing in an extension of **'wav'** instead of **'pat'**, will save an ascii-file with extension **.DES** containing descriptions of loop points and such for the instrument, as well as saving all the instrument's samples in **.WAV** files (name.des + namenn.wav).
- 5) Repeat from point 2) or 3) or select **Quit**.
- 6) Don't forget to Register as a user of this program.

✘

Only 16 characters can be stored in the.PAT file although more may be shown here.

Total size in Bytes for all waveforms in the instrument.  
Approximately the minimum amount of GUS memory needed.

Applies to the last wavesample in multisample instruments.

Total no of wavesamples in the (multisampled) instrument.

## File Formats

x

*2PAT can read the following file formats:*

<b><u>AIF</u></b>	Audio Interchange File Format (amiga & mac).
<b><u>AU</u></b>	Sun/NeXT audio files.
<b><u>EFE</u></b>	Ensoniq EPS 16+ disk image extract files. Same as INS.
<b><u>FSM</u></b>	Farandoyle Sample format.
<b><u>IFF</u></b>	Interchange file format (Amiga).
<b><u>INS</u></b>	Ensoniq EPS 16+ instrument files (extract from GKH files).
<b><u>KRZ</u></b>	Kurzweil 2000 sample files (also split files: KR1, KR2).
<b><u>MOD</u></b>	Amiga (4ch) modules and PC 4, 6, 8 chs.
<b><u>MTM</u></b>	MultiTracker Modules.
<b><u>PAT</u></b>	Ultrasound Patch files.
<b><u>RAW</u></b>	Raw signed PCM data.
<b><u>SDK</u></b>	Roland S-550, S-50, S-330 disk images.
<b><u>SDS</u></b>	Raw Midi Sample Dump Standard files.
<b><u>SDX</u></b>	Midi Sample Dump Standard files as saved by the program SDX.
<b><u>SMP</u></b>	Samplevision files.
<b><u>SND</u></b>	Raw unsigned PCM data.
<b><u>STM</u></b>	Scream tracker v2 modules.
<b><u>S3M</u></b>	Scream tracker v3 modules.
<b><u>SYW</u></b>	Yamaha SY-series wave files (really named W??).
<b><u>TXW</u></b>	Yamaha TX16W wave files (really named .W??).
<b><u>669</u></b>	669 tracker modules.
<b><u>ULT</u></b>	UltraTracker Modules.
<b><u>UWF</u></b>	UltraTracker Wave Files.
<b><u>VOC</u></b>	Creative labs sound files.
<b><u>WAV</u></b>	Windows PCM wave files.

*and save them as Gravis Ultrasound Patch files or as Microsoft Windows Wave files.*

x



*669 files: 669 tracker modules (PC)*

*Features:*

8 bits, looping, instrument name.

***Au files: Audio files (Sun/NeXT)***

*Features:*

8, 16 bits.

*Unsupported features:*

Multichannel files.

U-law encoding, other bitdepths (incl. floatingpoint...).

***AIF-files:      Audio Interchange File Format (Amiga, Atari, SGI)***

Multisampled instruments are stored in multiple files; as with SDS you must manually recombine them.

***Features:***

8, 16 bits, looping, keyrange.

***Unsupported features:***

Multichannel files.

Other bits/sample.

***Internet source of samples:***

*ftp.reed.edu* under */eps/samples/aiff*.

*FSM files: Farandoye wave files (PC)*

*Features:*

8, 16 bits, looping, instrument name.

***IFF files: Interchange File Format: 8SVX data (Amiga)***

***Features:***

8 bits, looping, name.

***Unsupported features:***

Multichannel files.

***Ins and Efe files:      Ensoniq EPS family instrument extract files***

The Ensoniq EPS (and family) files are usually encountered in form of disk image files called .GKH. You must extract the individual instrument files to .INS or .EFE files with a utility called EPSDISK in order to be import them in 2PAT. This utility can be found at *ftp.reed.edu* under *eps/utls*. INS files are the same as an EFE files but the latter may also contain EPS sequenced songs, bank info or such, instead of an instrument.

Multilayered instrument will have each layer shown as a separate instrument (all have the same name tough), Note! Sometimes there will be layers without wave-data; they will sound garbled or like static noise; just ignore them...

***Features:***

16 bits, multisampled, looping, wavename, instrument name.

***Unsupported features:***

Envelopes et c.

***Internet source of samples:***

*ftp.reed.edu* under */eps/samples* (stored in GKH format).

### *Krz files: Kurtzweil K2000 instrument files*

The format of Kurtzweil K2000 files seems to be 'secret' so any resemblance to the .Krz file format that this program can read are purely incidental :-/

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 :-), sometimes things may go a bit wrong here and there, but in most cases the result is fairly good. The worst problem at the moment is that no sample-rate information are currently converted so some manual tuning may be in order to please sensitive ears.

#### *Features:*

16 bits, multisampled, looping, wavename, instrument name.

#### *Unsupported features:*

Envelopes et c.

Only one channel of stereo instruments will be used.

If anyone want's to try to delve deeper into the Krz format, I'll will be happy to email you a description of what I've been able to find out.

#### *Internet source of samples:*

*ftp.uwp.edu* under */pub/music/lists/kurzweil/sounds*.

***Mod files:     Module files (Amiga, PC)***

Recognizes the following flavours:

M.K., M!K!, 4CHN, 6CHN, 8CHN, FLT4, FLT8 and OCTA.

***Features:***

8 bits, looping, instrument name.



*Mtm files: Mtm tracker modules (PC)*

*Features:*

8, 16 bits, looping, instrument name.

*Pat files: Gravis Ultrasound GF1 patch files (PC)*

*Features:*

8,16 bits, multisampled, looping, envelopes, tremolo, vibrato, pitch scaling wave name, instrument name, note.

*Raw files: Raw signed PCM data files*

Assumes 8 bits, 11KHz.

***Sdk files: Roland S-550, S-50, S-330 disk image files***

They might go under another extension but SDK seems to be most common; rename them into *.SDK* and 2PAT will recognize them. As with Sds files you must manually reassemble multisampled instruments.

***Features:***

12 bits, looping, name.

***Unsupported features:***

A zillion different parameters.

***Internet source of samples:***

*lotus.uwaterloo.ca* under */pub/sgroup/samples*.

*Sds & Sdx files:            Midi Sample Dump Standard files*

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

These files contain one sample each and multisampled instruments are saved in multiple files, you can save these as wave files and import them into a patch editor and manually restore loop points, key ranges and names from the info in the accompanying *.DES* file.

*Features:*

8..16 bits, looping. SDX only: wave name, note.

*Unsupported features:*

More than 16 bits.

*Internet source of samples:*

*sweaty.palm.cri.nz* under */sds/samples*.

*alf.uib.no* under */pub/midi/sds/samples*

*Smp files:      Samplevision files (PC)*

*Features:*

16 bits, looping, name, note.

*Unsupported features:*

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

*Snd files: Raw unsigned PCM data files*

Assumes 8bits, 11KHz.

*Stm files: ScreamTracker module files (PC)*

*Features:*

8 bits, looping, instrument name.



*S3m files:      ScreamTracker v3 module files (PC)*

*Features:*

8 bits, looping, instrument name, note.

### *Syw files: Yamaha SY-series wave files*

The Yamaha SY-series (or at least the SY-85 and SY-99) wave sample files are really named *.W??* but you must rename them into *.SYW* in order for 2PAT to recognize them; just do a "ren \*.w?? \*.syw".

There's also *.t??* (All Data Files, often contains waves and much more common than *.w??*; but aren't supported at the moment), *.j??* (Patch parameters?) and *.k??* (Sequenced songs).

As with Sds files you must manually reassemble multisampled instruments.

### *Note!*

This has not been tested; please send me some files if you have any! Information on the all-data format (*.t??* files) would also be appreciated.

### *Features:*

16 bits, looping.

### *Txw files: Yamaha TX16W wave files*

The Yamaha TX16W wave sample files are really named *.W??* (where ?? is a number) but you must rename them into *.TWX* in order for 2PAT to recognize them; just do a "ren \*.w?? \*.txw". 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.

As with Sds files you must manually reassemble multisampled instruments.

### *Features:*

12 bits, looping.

### *Internet source of samples:*

*ftp-ls7.informatik.uni-dortmund.de* under */pub/tx16w/samples*.

*Ult files: UltraTracker module files (PC)*

*Features:*

8,16 bits, looping, instrument name.

*Uwf files: UltraTracker wave files (PC)*

Not tested with 8bit files.

*Features:*

8,16 bits, looping, instrument name.

*Voc files: Creative Labs sample files (PC)*

*Features:*

8 bits, name.

*Wav files: Microsoft Wave files (PC, Windows, OS/2, NT)*

*Features:*

8,16 bits PCM.

*Unsupported features:*

Other bits/sample.

Stereo files.

Compressed files.

## Notes

x

### *General:*

The envelopes used for wave samples corresponds to the ones that Patch Maker use for looped and unlooped waves, except that while looped samples have a quite long decay time, it isn't as long as the long decay setting in Patch Maker. You may want to edit the envelope in any patch editor that supports it (i.e. PED at the moment).

### *Midi thru:*

If you can't get midithru to work (in order to play the samples), check:

- Have you connected your midi in device properly?
- Are your ULTRADIR= environment variable correct?
- Are any other (conflicting) program using the input device?
- The first free device is used, i.e. if no other program is using midi in it's the one under MIDI= in SYSTEM.INI (as opposed to MIDI1= et c.).
- If you don't have a midi in device, there's a program called PIANO that emulates one by providing an on-screen keyboard.
- If you can't hear a specific instrument but it normally work's it might be that it's a drum sound that's assigned to one key only, and you haven't found it, or it may be that it's too large to fit in GUS-memory..

### *Instrument size:*

Sometimes an instrument may be too large even for an 1MB GUS (there are people that have > 32MB on their K2000) especially since it can only use memory in banks of 256KB wich samples can't cross (so a worst case behaviour would be something like, if all the samples are 129KB, you could only four multisamples). In such cases you must use a patch editor to manually throw out samples until it fits (Note! if you're doing this with Patch Maker Lite you must save, close and reopen the patchfile, after removing a sample in order to hear (i.e. be able to play) it (if it now fits i.e.). Selecting 'Conserve memory, 8-bit' in the Ultrasound driver setup will let you load twice as much as well of getting rid of the 256KB limit (since it only applies to 16bit data); but then that degrades sound quality quite a bit...

### *Temporary files:*

2PAT will save temporary patches in the directory pointed to by the TEMP= environment variable if available, or else in the program directory.

### *Tracker module files:*

Patches converted from this category will have C4 as root frequency and 22.1KHz as sample frequency. If the sound was sampled at another frequency, you will have to adjust it yourself with a patch editor (there's no info about this in a typical tracker module file).



x

## Revision History

✕

### *future versions,*

More input formats; just send me a description of any format(s) you want me to include (or send me some files and I might reverse-engineer'em, I am especially interested in file formats for pro-sample synths and such...)

Win32 version; as soon as Windows 4 is out (not that many people have Windows 3.1 with Win32s installed today).

### **version 1.0, <94-04-27>**

Now also able to write WAV files in addition to PATches. Typing in an extension of '.wav' instead of '.pat' in the Save dialog box, will save a .DES text file with into about loop points and such, as well as .WAV files with the samples in the patch.

Converted the manual into WinHelp Hypertext format.

Reads INS and EFE files (Ensoniq EPS instrument files). These can extracted from GKH files with a utility called EPSDISK.

Reads SDS files (raw Midi Sample Dump Standard files).

Reads SDX files ('compacted' Sds file + sample name, as saved by the program SDX).

Reads TXW (Yamaha TX16W wave files). Really named .W??.

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

Has not been tested (Please send me some files!)

Note! The much more common all-data format, .T?? are not supported (yet).

Reads SDK (Roland S-550, S-50, S-330 disk images).

Reads AIF files (Audio Interchange File Format).

Reads IFF: 8SVX files (Interchange File Format).

Reads UWF files (UltraTracker Wave files).

Reads PAT files (Ultrasound patch files).

Reads SMP files (Samplevision).

Reads VOC (Creative labs) files.

Reads AU files (Sun/NeXT); only uncompressed mono files supported.

Reads RAW signed data files; assumed 8bit, 11KHz.

KRZ files now get their loop point. Now also supports split files (KR1 + KR2). No longer throws out samples if the instrument is too big but rather lets you do it yourself in the patch editor of your choice.

Correctly reads 16bit WAV files now.

Correctly reads ULT's with 16bit samples now.

Looped samples with the end loop point at or near the sample end now get the sample at the loop start point duplicated at the loop end point + 1, in order to keep the GUS interpolation happy (can cause clicks otherwise).

The envelopes now corresponds to the ones Patchmaker use for looped and unlooped waves, except that looped samples have a quite long decay but not at all as long as the long decay setting in Patchmaker.

Now using FMJULTRA.DLL to provide ULTRASND.INI manipulation for loading custom patches; email me if you'd like to use it in your own programs and I'll send you FMJULTRA.H.

Renamed MIDITHRU.DLL into FMJMTHRU.DLL. In order to avoid possible conflicts with other programs.

Various minor changes and improvements.

#### **version 0.9B, <94-03-30>**

Changed program name to 2PAT (from MOD2PAT, after all it reads a lot more stuff than just MOD files).

Reads Kurzweil K2000 sample files (.KRZ); multisampled supported. Since the spec. is 'secret' I have only been able to find out where the actual samples are stored, how long they are, what pitch and what keys they are assigned to and names and such, but haven't been able to determine loop points, envelopes et c.

Reads .SND, raw unsigned data files. Assumes they are 11KHz, 8bit.

Hopefully the problem with samples > 64KB is fixed.

Saves temporary patch-files in the directory specified by the TEMP= environment variable or, if not available, in the program directory.

Faster updating of ULTRASND.INI.

Nicer look with BWCC.

**version 0.8B, <94-03-08>**

Now a Windows hosted program (Win16).  
The user interface is naturally quite different in this incarnation...

The .INI extract feature is no longer implemented, nor is reading raw files.

Patch play feature. If you've got an Ultrasound card (and you have, haven't you?) and a midi in device, you can play the patches from within mod2pat.

**version 0.7B, <94-01-07>**

Removed the signed to unsigned conversion.  
Since we've now got Patch Maker Lite, wich hasn't got the bug of the old PATCH.EXE.

The .INI extract now looks a bit differently.

I've introduced the concept of "BankName=Whatever" to give the banks names; it will be set to the songname.

Added my bird logo to the text font.

**version 0.6B, <94-01-05>**

Fixed another embarrassing bug, when converting from signed to unsigned data I now add 80h and not 7Fh as previously (got a little noise...).

**version 0.5B, <93-12-23>**

Yes, now you can 'hear-out' the samples before you decide wether to save them. Of course, an ultrasound card is required for that!

Fixed a rather embarrassing bug with MOD loop points.

**version 0.4B, <93-12-20>**

Reads 669, STM and S3M.

Always saves as unsigned, 'this because Gravis PATCH.EXE always uses unsigned data and it's convert command doesn't convert the last byte...

Converted source code into C.

**version 0.3B, <93-11-26>**

Reads MTM, ULT, WAV, FSM and raw files.

Ability to edit internal patch names (Use TAB to toggle between DOS-name and Patch-name edit).

Added GUSPATCH.TXT; Ultrasound patch format description.

Any MOD that isn't a M.K., 4CHN, 6CHN, 8CHN, FLT4 or FLT8 are now treated as being of the older 15 (versus 31) sample format.

**version 0.2B, <93-11-21>**

Reads 4,6,8 ch MOD's 'M.K.', '4CHN', '8CHN'.

Basic user interface.

Smart 'auto filename'.

**version 0.1B, <93-11-13>**

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

✘

## Adds

x

### *Midi Patch Browser (v0.82)*

A generic midi integrator that'll give you a friendly interface for easily selecting and playing patches on your midi system.

Fully supports multiple patch bank caching as well as GS banks.

Now also has special Ultrasound support for playing any .PAT file on disk.

[MPB082.ZIP]

### *Ultrasound Patch Librarian: (v0.1)*

Load and manage arbitrary patches from disk to GUS-memory.

Let's you load and save lists of patches.

[UPL01.ZIP]

### *Image Eye (v2.01)*

You'll not be able to find a faster GIF-viewer.

Handles a lot of formats incl. JPEG with good quality conversion to 256clrs (from 16M). Requires i386+, XMS, VESA.

[IEYE201.ZIP]

x

