Protrekkr v2.5.5

Operating Manual

Written by Franck Charlet (charlet.franck@wanadoo.fr)

Introduction

ProTrekkr (formerly known as NoiseTrekker by Juan Antonio Arguelles Rius aka Arguru) is a tracker program combining a software synthesizer together with a traditional samples tracker which can (mainly) be used to create electronic music (like psytrance, trance goa, hard acid, IDM, chip, techno, jungle, etc.) for small sized intros, demos or games.

The tracker is available for Windows, Linux, Mac OS X, FreeBSD, Amiga OS4, AROS & Haiku OS.

Main features:

- 16 multi-notes tracks (with 256 virtual channels).
- Up to 4 effects per track.
- Samples editor (can load up to 64 bits stereo samples).
- Patterns based tracker style sequencer.
- Synthesizer with custom waveforms support.
- Samples loop editor.
- 2 303s units.
- Cubic spline interpolation.
- Wav files import/export.
- 16 or 32 bits wav rendering.
- Multi effects engine with reverb, distortion, filters, delay, EQs, flanger, etc.
- Fully customizable 10 combs reverb.
- Midi in/out support.
- Protracker .mod files import.
- Samples compressor handling Gsm/Mp3/TrueSpeech/ADPCM/8 Bit/WavPack (only on Windows).
- Replay routine source code to be included & used inside your own programs (Also featuring a replay routine for the PlayStation Portable).
- Winamp/XMPlay plugin available.
- And maybe more...

This is a FREEWARE tool, this means that you can use it for whatever you want and for free. There's NO GUARANTEE that this program will perform well or even run on your computer.

Minimum requirements

The tracker can be quite CPU intensive (especially when the CPU have to handle visuals as well but nowadays most of that is done on GPUs), a 1.5ghz processor may be the bare minimum when replaying heavy modules (16 tracks all effects/polyphony on).

The only computer running under Mac OS X i have is a dual 450Mhz G4 and it doesn't make it at all, there's a lot of hiccups (although the system monitor doesn't show more than 60% usage for both CPU so i may have made a mistake somewhere).

How to install (Windows only)

On Windows, go to the install directory, right click on the "LameACM.inf" file and selected "install". Do the same for the "atrac3.inf" file, this will install both codecs in the system. Without these 2 codecs installed, ptk wouldn't be able to pack samples in Mp3 or Atrac3 (only used on PSP) formats.

Compressed modules (.ptp)

These modules can be recognized as they're using the ".ptp" extension (regular modules use ".ptk"), they can’t be reloaded inside the tracker, but the musician can simulate the compression of the samples (or per sample) within the tracker.

While these files may have a bigger size than their .ptk counterparts, they're internally organized to maximize packing ratio so using any generic data compressor on them will produce a much smaller file than what could be achieved by packing a .ptk file.

There are two commands related to compressed modules in the tracker:

- "**Save .ptp**":
  Create a module with extension ".ptp" in the current directory.

- "**Calc .ptp size**":
  Calculate (roughly) the size that the module would take once used in your program.

Note that compressed modules generated on big endian platforms and little endian ones aren't compatible between them.

Samples inside .ptp modules can be packed, the user have the ability to select the packing algorithm which will be used on a per sample basis, the available compression schemes are all “lossy”.

WavPac:
Wavpack based compression (lossy algorithm).

GSM:
Suitable for drums/snares.

MP3:
Suitable for longer samples like cymbals.

ADPCM:
Can give better result than GSM (but samples are a bit bigger).

At3 (For PSP):
This one is similar to MP3 compression, it will be handled in the PSP replay routine only (but it's not handled yet).

8 Bit:
The samples (every 2 samples in fact) are packed to 8 bits and interpolated to 16 bits when depacked this give a bit better results than adpcm but compress less.

None:
Suitable for very short (chiptune like) samples (if you want to keep quality).
Note that this is the only way to save packed samples if the tracker has been compiled without codecs supports.

The .ptp are the only modules format recognized by the replay routine.

Note:
The MP3 compression might not work on a bare bone Windows 2000 (it works on any others).

Note2:
When a packing is selected for a sample (other than none) samples are reconstructed by interpolation.

Alongside with the .ptp are the optional .psm files which contain synchro informations related to the effect "07" used within the patterns; The .psm file format is as follow:

Header:

1.w: number of synchros data entries

Synchro data:

1.b: position
1.b: row
1.b: fx data (as specified within the tracker)

Note:
The fx data part is removed from the .ptp files and moved to the .psm files and only one fx per position/row is saved.

FAQ/Troubleshooting

Q> No Fasttracker support, no 53194 tracks ? No etc...?, "m... i dont like the gui, the key shortcuts, you should have to..."

A1> Get a compiler and code the best tracker ever. I'm sure you'll do fine.
A2> Use fastracker, impulse tracker, buzz or whatever stuff you want.
A3> Use your browser and search for some stuff...

Q> I Cannot run the program! Any idea ?

A> If u have got a SBLive, try disabling EMU driver on your soundcard. Be sure you have got lastest DirectX drivers for your stuff (gfx/sfx card) and DirectX6.1 or above installed. (Microsoft DirectX 6.1 or above is required to run ptk) Ptk won't run on Windows NT, it should run on any other Windows. Also, might not work with some goofy gfx/sfx cards.

Q> Well, the program runs ok but ... how I can quit it ?

A> Click 2 times on the exit button. Or just close the window if you're running ptk in windows mode. Alternatively, you can also use the key combo: LALT + F4.

Q> Ehh Everything sounds clicks, distorted and weird !!?

Plan A> Try to use higher latency in Misc. Setup section. This means that the CPU is not having enough time to render/fill the audio buffer data, and this produces clicking…
Plan B> Buy a faster computer.

Q> How I play 303 patterns in ptk ?

A> Use ptk command 31xy and 32xy. 31xy will trigger pattern 'y' (1-8) from bank 'x' (A-D) from the first 303 unit (ptk has got 2 303 units, just like Rebirth).

Example:

```
--- .. .. .. 31A3....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 0000....
Off .. .. .. 0000....
```

The 303 unit 1 will be assigned to the track 0 stream.

This means, that panning, fx setup, filter settings of the track will be applied to the sound of the 303 unit 1.

To play the second 303 unit, use the same method but with 32xy command instead of 31. To stop 303 playing (patterns are played "looped" continuously) just put a note off (right shift key) on the track were it was triggered.

Also, using -3100- or -3200- will replay the patterns that are currently selected on the 303 editor. Can be handy while composing basslines.

Btw: 303 engine will not "eat" the track sampler/csynth engine, so you can have both playing ex. a bassdrum and a 303 line on the same track.

Also, both 303 units can be triggered in the same track. Well, just play with them :].

Q> Oh!, the 303 sounds great, will you code a 'buzz' port ?

A> Nope.

Q> How I can record 303 or the tracks tweakings on the fly ?

A> Press the "Live Rec: OFF" (top/left of the screen) button and, voila, the 303 tweakings or current track are auto-filled on pattern while the tune is being played.

Q> How I can alter 303s CutOff, Resonance, etc.. on the pattern while playing ?

A> Easy, just use the 303 special pattern commands. They're very useful to automatize 303 stuff.

Q> How to use Midi In/Out ?

A> First of all, midi in & out capabilities of this program are rather “primitive” to say the least.

Follow these steps:

1 - Go to Misc. Setup section and select a midi in or out device to use (ptk only supports one device at a time).

2 - Go to instrument section, and select a MIDI PRG (the default is N/A, which means no midi program selected).

3 - Go to track section and here u can assign a midi channel to each track of ptk.

4 - Play notes :]. Note off works. F'x' note cut command also works too, and note-volume command (speed) is supported.

Also, you can change midi controllers in the tracker, using '90' in the panning row:

ex:

```
C-3 02 .. .. 0000....
--- .. .. 90 xxyy....
--- .. .. .. 0000....   of the controller n.'xx' to 'yy' (both in hex)
--- .. .. .. 0000....
```

So:

```
--- .. .. 90 2040....   will set the controller number $20(32) to value $40(64).
```

You will need the midi implementation table of your gear to know what you can change with midi controller messages. Probably, it's located at the end of the manual =].

Q> Audio & Midi are not synchronized, what I can do ?

A1> Buy a commercial software package.
A2> Well, there is a nasty trick to synchronize both. It's a bit hardcore but work with me:

Simply put one line down to all midi notes on your pattern (use Insert key) and go to 'Misc. Setup', adjust the latency and just search a value that will make sound sync both audio/midi.

Q> The stock Sin/Saw/Pulse and Rnd waveforms are too simple/common, is there a way to use something more complex/rich ?

A> You have to ability to redirect the waveforms of the instruments through the synth pipe by selecting the "wav" option for the oscillator you're using for

this synth instrument, samples can be used as wavetables to replace the stock signals.

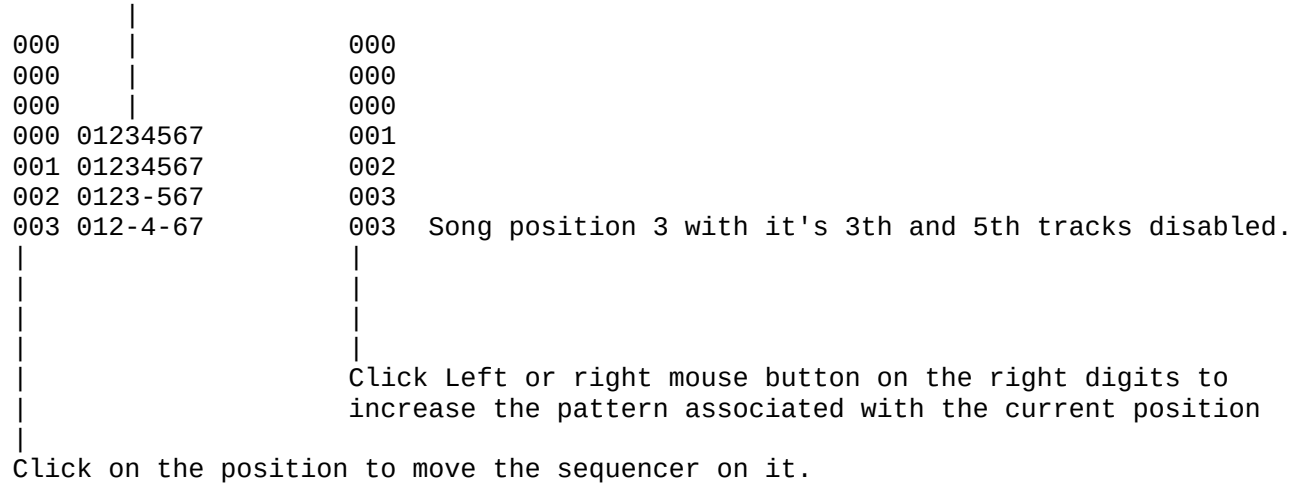
Q> Will you support VST instruments in a future?

A> Yes, as specified here http://code.google.com/p/protrekkr/issues/list but that's not really a priority, the main goal of the tracker is to be a all-in-one program with a compact & re-usable replay routine.

The sequencer

The sequencer used in ptk is a little bit different from those usually found in other trackers, the user have the ability to re-use patterns with different tracks playing state configurations:

Tracks can be turned on/off by left clicking on their digits. (Inactive tracks will be displayed with a darker font color). (Right clicking will solo/unmute all tracks alternatively).



(The mouse wheel can also be used to scroll through positions sequence).

The effects

Command '**0000**': **No Effect**

Command '**01xx**': **Pitch Up** (Disabled when using Arpeggio or Vibrato)

xx = Speed

Command '**02xx**': **Pitch Down** (Disabled when using Arpeggio or Vibrato)

xx = Speed

Command '**03xx**': **Set Volume**

xx = Volume level

Command '**04xx**': **Trance Slicer**

xx = Delay ticks before muting the volume (from 0 to 6) (note that the volume stays muted at the next rows)

Command '**05xx**': **Glider** (Disabled when using Arpeggio or Vibrato)

xx = Speed

Command '**06xx**': **Pattern Loop**

00 = Sets the loop point
xx = Loops xx times

Command '**07xx**': **FX Synchro**

xx = Any data

Command '**08xx**': **Set Filter Cut Off**

xx = Cut Off level

Command '**09xx**': **Set Sample Play Offset** (256 bytes steps)

xx = Offset

Command '**0Axx**': **Randomize Filter Cut Off**

xx = Amplitude

Command '**0Bxx**': **Filter Cut Off Slide Up**

xx = Speed

Command '**0Cxx**': **Filter Cut Off Slide Down**

xx = Speed

Command '**0Dxx**': Jump To Next Sequencer Position (aka Pattern Break)

xx = Pattern row

Command '**0Exx**': **Note Retrigger**

xx = Tick

Command '**0Fxx**': **Set Number Of Ticks Per Beat**

xx = Ticks

Command '**10xx**': **Set Delay/Echo Send**

xx = Send Level

Command '**11xx**': **Set Reverb Send**

xx = Send Level

Command '**12xx**': **Set Distortion Threshold**

xx = Threshold Level

Command '**13xx**': **Set Distortion Clamp**

xx = Clamp Level

Command '**14xx**': **Set Filter Resonance**

xx = Resonance Level

Command '**15xx**': **Set Filter Type**

xx = Filter to use

00 = LoPass -12db filter
01 = HiPass filter
02 = BandPass filter
03 = BandReject filter
04 = No Filter
05 = LoPass -24db filter
06 = LoPass -48db filter
07 = LP-24 [Stereo] filter
08 = A. Mod. [Mono] filter
09 = A. Mod. [Stereo] filter
0A = Single [Mono] filter
0B = Single [Stereo] filter
0C = ParaEq -15db filter
0D = ParaEq -6db filter
0E = ParaEq +6db filter
0F = ParaEq +15db filter
10 = Custom Delta filter
11 = Comp. Distort L filter
12 = Comp. Distort M filter
13 = Comp. Distort H filter
14 = Comp. Distort filter
15 = W-HP12 [Mono] filter
16 = W-HP12 [Stereo] filter
17 = W-HP24 [Mono] filter

Command '**16xx**': **Set Filter LFO Carrier position**

xx = Position

Command '**17xx**': **Auto Fade In Track**

xx = Ticks

```
C-7 00 .. .. 1710....
--- .. .. .. 0000....
--- .. .. .. 0000....
G-6 00 .. .. 0000....
--- .. .. .. 0000....    and will be faded in too
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 1700....
```

Command '**18xx**': **Auto Fade Out Track**

xx = Ticks

Same behavior as above.

Command '**19xx**': **Volume Slide Up**

xx = Speed

Command '**1Axx**': **Volume Slide Down**

xx = Speed

Command '**1Bxy**': **Arpeggio**

x = Second relative semi tone
y = Third relative semi tone

```
C-7 00 .. .. 1B37....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 1B38....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 1B00....
```

Command '**1Cxx**': **Set Global Volume**

xx = Volume level

Command '**1Dxy**': **Vibrato**

x = Speed
y = Depth

```
C-7 00 .. .. 1D46....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 1D7F....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 0000....
--- .. .. .. 1D00....
```

Command '**1E0x**': **Select Instrument Playing Way**

0 = Play forward
1 = Play backward

Command '**1Fxx**': **Position Jump**

xx = New position

Command '**20xx**': **Fine Volume Slide Up**

xx = Amount

Command '**21xx**': **Fine Volume Slide Down**

xx = Amount

Command '**22xx**': **Fine Pitch Up**

xx = Amount

Command '**23xx**': **Fine Pitch Down**

xx = Amount

Command '**240x**': **Turn flanger on/off**

0 = Turn it off
1 = Turn it on

Command '**25xx**': **Set shuffle value**

xx = Amount

Command '**26xx**': **Set reverb filter cutoff**

xx = Amount

Command '**27xx**': **Set reverb filter resonance**

xx = Amount

Command '**280x**': **Turn track filter on/off**

0 = Turn it off
1 = Turn it on

Command '**F0xx**': **Set BPM Speed**

xx = BPM

Command '**31xy**': **Trigger 303 Bass Line (Unit 1)**

x = Bank (A-D)
y = Pattern (1-8)

'00' will trigger current selected pattern
‘ff’ will stop the bass line

Command '**32xy**': **Trigger 303 Bass Line (Unit 2)**

x = Bank (A-D)
y = Pattern (1-8)

'00' will trigger current selected pattern
‘ff’ will stop the bass line

Where 'xx' is a number between $00 and $FF (hex):

Command '**33xx**': **Set 303 (Unit 1) Filter Cutoff**
Command '**34xx**': **Set 303 (Unit 2) Filter Cutoff**
Command '**35xx**': **Set 303 (Unit 1) Filter Resonance**
Command '**36xx**': **Set 303 (Unit 2) Filter Resonance**
Command '**37xx**': **Set 303 (Unit 1) Filter Env Mod**
Command '**38xx**': **Set 303 (Unit 2) Filter Env Mod**
Command '**39xx**': **Set 303 (Unit 1) Filter Decay**
Command '**3Axx**': **Set 303 (Unit 2) Filter Decay**
Command '**3Bxx**': **Set 303 (Unit 1) Accent**
Command '**3Cxx**': **Set 303 (Unit 2) Accent**
Command '**3Dxx**': **Set 303 (Unit 1) Tune**
Command '**3Exx**': **Set 303 (Unit 2) Tune**
Command '**41xx**': **Set 303 (Unit 1) Volume**
Command '**42xx**': **Set 303 (Unit 2) Volume**

The volume column

Command '**xx**': **Set Volume (00 to 40)**

Command '**Fx**': N**ote Cut**

x = tick number where the note should be cut

The panning column

Command '**xx**': **Panning (00 to 80)**

Command '**90**': **Send Midi message**

Send the command and data specified in the fx/data columns to Midi OUT selected device.

ex: 90 4050

Send command 0x40 with data 0x50.

Midi command:

Command '**80xx**': **Set Patch Bank**

(might not work on all midi equipments)
(**won't work in the replay routine**)

Read in the [faq] how you can send midiout controller messages and read the manual of your gear to learn how send RRPN ones using midiout, also, you might find here the midi implementation table for your hardware (or it could also be displayed on the device's LCD screen).

The shortcuts

Playing

RCTRL:	Play song from row 0.
LSHIFT + RCTRL:	Play song from current row.
RALT:	Play pattern from row 0.
LSHIFT + RALT:	Play pattern from current row.
Left mouse on '>':	Play song from row 0.
Right mouse on '>':	Play song from current row.
Left mouse on '|>':	Play pattern from row 0.
Right mouse on '|>':	Play pattern from current row.
Left mouse on 'Edit/Record':	Edit mode on/off.
Right mouse on 'Edit/Record':	Record mode on/off.

Editing

LSHIFT + ESCAPE:	Switch large patterns view on/off
TAB:	Go to next track
LSHIFT + TAB:	Go to prev. track
LCTRL + TAB:	Go to next note in track
LCTRL + LSHIFT + TAB:	Go to prev. note in track
SPACE:	Toggle Edit mode On & Off (Also stop if the song is being played)
SHIFT SPACE:	Toggle Record mode On & Off (Wait for a key note to be pressed or a midi in message to be received)
OWN ARROW:	1 Line down
UP ARROW:	1 Line up
LEFT ARROW:	1 Row left
RIGHT ARROW:	1 Row right
PREV. PAGE:	16 Arrows Up
NEXT PAGE:	16 Arrows Down
HOME / END:	Top left / Bottom right of pattern
LCTRL + HOME / END:	First / last track
F5, F6, F7, F8, F9:	Jump to 0, 1/4, 2/4, 3/4, 4/4 lines of the patterns
+ - (Numeric keypad):	Next / Previous pattern
LCTRL + LEFT / RIGHT:	Next / Previous pattern
LCTRL + LALT + LEFT / RIGHT:	Next / Previous position
LALT + LEFT / RIGHT:	Next / Previous instrument
LSHIFT + M:	Toggle mute state of the current channel
LCTRL + LSHIFT + M:	Solo the current track / Unmute all
LSHIFT + F1 to F11:	Select a tab/panel
LCTRL + 1 to 4:	Select a copy buffer

Tracking

1st and 2nd keys rows:	Upper octave row
3rd and 4th keys rows:	Lower octave row
RSHIFT:	Insert a note off
/ and * (Numeric keypad) or F1 / F2:	-1 or +1 octave
LCTRL + F1 / F2:	Decrease / Increase editing step.
F3:	The chords helper is triggered when the caret is located on a note.
INSERT / BACKSPACE:	Insert or Delete a line in current track or current selected block.
LSHIFT + INSERT / BACKSPACE:	Insert or Delete a line in current pattern
DELETE (NOT BACKSPACE):	Empty a column or a selected block.

Block manipulations

(Blocks can also be selected with the mouse by holding the right button and scrolling the pattern with the mouse wheel).

```
LCTRL + A:          Select entire current track
LCTRL + LSHIFT + A: Select entire current pattern
LALT + A:           Select entire column note in a track
LALT + LSHIFT + A:  Select all notes of a track
LCTRL + X:          Cut the selected block and copy it into the block-buffer
LCTRL + C:          Copy the selected block into the block-buffer
LCTRL + V:          Paste the data from the block buffer into the pattern

LCTRL + I:          Interpolate selected data from the first
                    to the last row of a selection
```

example: (assuming only the 2 effects data columns are selected)

```
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  00
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  02
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  05
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  08
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  0A
C-3  04  ..  ..  09  00 ---> C-3  04  ..  ..  09  0D
C-3  04  ..  ..  09  10 ---> C-3  04  ..  ..  09  10
```

Cool to make cutoff transitions, etc...
(You can also switch on the Slider Rec to On, and perform parameter-live-recording, such as cutoff, resonance or panning tweaking, etc..)

Note:
this command (as well as the randomize one below) only works for volume/panning and fx data columns.

```
LSHIFT + ARROWS
PREV. PAGE
NEXT PAGE:          Select a block

LCTRL + R:          Randomize the select columns of a selection, works similar
                    to CTRL + I, but it randomizes values instead of
                    interpolating them.

LCTRL + U:          Transpose the note of a selection to 1 seminote higher
LCTRL + D:          Transpose the note of a selection to 1 seminote lower
LCTRL + LSHIFT + U: Transpose the note of a selection to 1 seminote higher
                    (only for the current instrument)
LCTRL + LSHIFT + D: Transpose the note of a selection to 1 seminote lower
                    (only for the current instrument)
LCTRL + H:          Transpose the note of a selection to 1 octave higher
LCTRL + L:          Transpose the note of a selection to 1 octave lower
LCTRL + LSHIFT + H: Transpose the note of a selection to 1 octave higher
                    (only for the current instrument)
LCTRL + LSHIFT + L: Transpose the note of a selection to 1 octave lower
                    (only for the current instrument)
LCTRL + W:          Save the current selection into a file
```

Misc.

```
LALT + ENTER:       Switch between full screen / windowed mode
LALT + F4:          Exit program (Windows only)
LCTRL + S:          Save current module
LSHIFT + S:         Switch top right panel to synths list
LSHIFT + I:         Switch top right panel to instruments list
```

Keyboard definitions

In order to keep a constant layout for the keyboard, the tracker uses locale keyboards text definitions which are located within the skins directory.

These files contains the full name of the region on the first line and the 37 ASCII codes used for that particular type of keyboard (the keys layout) following on 4 rows.

To add a new keyboard definition:

first copy an already existing file and rename it after the keyboard's country it is supposed to describe, modify the 4 rows then edit the skin.xml file and insert it's file's name into the value key of the "keyboards" tag along with the others.

Load (or reload) the tracker and select the new keyboard in the “UI Setup” panel.

Hexadecimal notation is allowed in the form 0xXXXX or just plain ASCII.
(it *may* work with unicode characters but i haven't tested it).

You can eventually submit a new keyboard layout at: charlet.franck@wanadoo.fr

Compiling the source

Compiling the tracker on an already supported operating system will require the 3 src/extralibs (that is SDL_draw, tinyxml & zlib-1.2.3) to be generated first, only the tracker itself can be generated with either one of the makefiles of the top directory or via the visual studio project.

In order to use the replay routine either use the provided makefiles or vc studio projects located in the release/distrib/replay directory, a .ptp module file and a ptk_properties.h source file (both generated by the tracker). Replace the PTK_MODULE variable content of the assembly files located in release/distrib/replay/test with the filename of the .ptp module and compile.

Porting the tracker

In order to port the tracker to another operating system, several issues should be considered (beside crafting the relevant makefiles, that is):

1. The keyboard handling:

It is important for the keyboard to behave identically on every port. The input is usually handled differently depending on the computer or even the operating system, for example the Windows version doesn't require keyboard definition files but the Linux one does and on Mac OS X the translation to unicode is handled differently.

2. The executable path:

The variable "ExePath" must be filled with the current path of the executable file during main(), all the data loading are relative to this variable.

The way to obtain it can be different depending on the operating system.

3. The midi drivers:

These can be disabled at compilation by using the __NOMIDI__ constant, the midi in/out drivers are located in src/midi/midi_drv_xxxxx.cpp files.

The functions of the drivers mainly concern enumerating the devices, opening them, receiving/sending messages and calling the callback function for incoming messages.

4. The codecs:

To reduce the size of the modules, the tracker is using (lossy) codecs present in the operating system to pack (and depack) the samples in .ptp module files. This can be disabled at compilation too by using the __NOCODEC__ constant. On operating systems which don’t possess codecs only Internal (wavpack) and 8 bit compressions are available.

Note that packing samples in .ptp files is not mandatory (the packing scheme for each sample can be set to none in the tracker).

Thanks

Karsten Obarski:

The father of the original tracker.

Arguru (RIP):

The original NoiseTrekker.

Fredrik Wikstrom:

Amiga OS 4 port.

Fernando Mastandrea:

AROS port.

Samuel Crow:

Haiku port.

Gary P. Scavone:

Rtmidi class.

Daxx, Teis, Kaneel, Ne7, Preacher, Mice, Kmuland, Manwe, Manbearpig, Misioslaw, Raina, Grusbanan, Syphus, Nula, Little bitchard, Lemmon, cp, matchugovsky, Evil, Mike West, Theorize, phecda, sk.syncsound:

Beta testing, ideas & bugs reports.

99:

Lunch with the gods.ptk

Archangel:

Wonderland.ptk

Bassie:

Gone tipsy.ptk

Bionic:

Factory of hybrid.ptk

DJ Amfibia:

Our trip to syrius.ptk

Doktor:

Aquarium.ptk

Dune:

Alterraid.ptk
Aulral.ptk
Dne_wro3.ptk

Drawer:

AcidTrance.ptk

Falcon:

Vacation climate.ptk
Without performer.ptk

Ganja:

Plastic elements.ptk

Jiffypop23:

Arrhenius.ptk

Nula:

Ordinary-Zero.ptk

Maktone:

Comic Bakery Remix.ptk

Mice:

Bitchbiker.ptk
Primitive.ptk

Okeanos:

Sudoku padawan.ptk

Revisq:

Goa.ptk
Trip to exp.ptk

Rhino:

Brass tacks.ptk
C-Masters.ptk
Chemistry.ptk
Dragon.ptk
Dream weaver.ptk
Jameson.ptk
Money.ptk
Round A Bout.ptk

Scorpik:

Love me daddy.ptk

Tomchi:

Devenirunpoulet.ptk

Wal:

Elevator zax.ptk
Loop influenza.ptk

Wally:

Space game.ptk

Xtd:

Flight.ptk
Warlock.ptk

Yzi:

Amarillo.ptk
Spinning.ptk

Michael Wolniak, AAS Tao, Arguru:

Presets and instruments.

Leonid:

303 skin.

Alien^PDX:

The startup logo.