



## At a glance

- [Introduction](#)
- [Why the name?](#)
- [What's new in this version?](#)
- [Will it run on a 486?](#)

## Operations

- [Playing the Mellosoftron live](#)
- [Using the Mellosoftron as a virtual MIDI device](#)
- [Editing an existing instrument patch \(PRG\)](#)
- [Creating a new instrument patch \(PRG\)](#)
- [MIDI input and MAPPING](#)
- [Controlling audio output](#)

## Additional resources

- [How to get more instruments](#)
- [How to create your own WAVs](#)
- [How to render MIDI files to WAV using Mellosoftron instruments](#)

## Ordering the Mellosoftron

- [Ordering overview](#)
- [Order online](#)
- [Order offline](#)

## For developers

- [Use the Mellosoftron engine in your own programs!](#)

## Getting tech support

Submit error reports and questions to [support@polyhedric.com](mailto:support@polyhedric.com).

**Before sending us a question, please make sure that it isn't answered in this document.** Don't forget to use the [Find](#) button - it's much faster than an e-mail query. Our own response time depends mainly on the number of redundant questions we have to handle.

## Introduction

The Mellosoftron is a low latency, fully programmable virtual sampler created by Tommy Anderberg and brought to you by Polyhedric Software.

In short, it turns your PC into a sampler which you can play live with a MIDI keyboard, with a sequencer or with any MIDI file player. Any 16 bit sound card with a DirectSound 3 (or later) driver will do - you don't need a particular brand or model. You can edit all instrument patches and create new ones from scratch using your own WAVs as sound sources.

Since it comes with an integrated patch editor and uses the same file formats (PRGs and WAVs) as the MIDnight Express and WAVmaker, the Mellosoftron is the ideal tool for the creation of your own instruments. Add the Acid WAV synthesizer and sound editor and you have a virtual recording studio inside your PC!

You can always download the latest versions of the Mellosoftron Evaluation Package and of related software from our site

[www.polyhedric.com/software](http://www.polyhedric.com/software)

## Why the name?

The Mellosoftron's name is a tribute to the grandfather of all samplers, the Mellotron.

This venerable ancestor was an electro-mechanical marvel. For each key on the console there was a magnetic tape player. When a key was pressed, the tape started rolling. Upon key release, a spring yanked the tape back (faster than a regular, motor-driven rewind, but still too slow for fast solo playing). Each tape contained a separate, 8-second recording - they had to be that long since no looping was possible with this approach.

The tapes came in racks which could be replaced to change "patches". Unfortunately, every time you did this, you were likely to also have to realign the tape player heads. For live performances, the Mellotron was therefore in practice a mono-timbral machine.

In spite of these limitations, the Mellotron's polyphony made it a success. It was the first keyboard capable of producing complicated chords with realistic-sounding ensemble instruments, in an age when electronic synthesizers were essentially monophonic beasts and digital samplers were unheard of. It was eventually pushed out of the market with the advent of electronic machines such as the ARP String Ensemble.

If you want to hear a Mellotron in action, listen to the chords in Jean Michel Jarre's first LPs, [Oxygene](#) (1976) and [Equinoxe](#) (1977).

The Mellosoftron can do a whole lot more than the old Mellotron - in fact, it could easily emulate one, given enough RAM - but version 1.x shared the latency problem which made its electro-mechanical ancestor unsuitable for fast solos. This echo of the past has faded away with the Mellosoftron II, but the name has stuck.

What's new in this version?

### **Short answer**

Just about everything.

### **Long answer**

The user interface has been completely redesigned and rewritten, exploiting the fact that ever faster CPUs and improved Windows resource management now allow more (and graphically heavier) controls to be kept on screen without severe performance effects:

- Most of the functions which used to reside in the MIDI dialog are now directly accessible from the main window. The MIDI dialog is now only used to select MIDI input device.
- The graphic keyboard has also been moved to the main window, together with "mousepad" controls which replace the old sliders and allow you to control two parameters at a time with the mouse (you can still remove keyboard and controllers from your sight simply by resizing the main window).
- The new PRG editor allows multiple selections and copy/paste operations. Even better, it has been integrated with the WAV viewer, making patch creation and editing much faster. No more switching between PRG and WAV windows: you can now edit sample positions the same way as envelope settings, both graphically and numerically.
- All windows are now resizable. The low resolution screens still found on some laptops are no longer a problem - even VGA will do.

And yes, it does look much cooler now!

There are also a couple of "hidden" user interface goodies for users who like to tinker:

- PRG filenames are now automatically translated to instrument names and displayed on the Mellosotron's instrument pads. Even better, the translation table is fully user-definable and can be edited with any text editor (e.g. Windows Notepad). The default table is in the file ML.INS in the Mellosotron's directory. You can override it by putting a file with the same name in the current PRG directory (this allows you to provide a unique instrument name table for each PRG directory you create). You can also edit the default table.
- The alphanumeric keyboard mapping is now user-definable. German-language users with QWERTZ-type keyboards and others who wish to change the notes assigned to each key can easily do so by editing the file ML.KCT in the Mellosotron's directory. This too is a plain text file.

Finally, no respectable techno freak can have missed this piece of news:

- The Mellosotron now has 16 MIDI-controlled resonant filters (one for each instrument pad), fully compatible with WAVmaker's filter implementation. Since filtering is computationally intensive, you can enable/disable each filter separately at any time. The MAP file format has been extended to include this information too, so the Mellosotron will remember which filters are active in your MAPs.

Will it run on a 486?

In principle, yes. In practice, a 486 is too slow for the Mellosotron. You need at least a Pentium 90 just to launch it, and an even faster CPU (as of version 3.0, at least a Pentium 266 is recommended) for reliable low latency operation. Please note that Pentium clones from Cyrix and AMD (prior to the Athlon) tend to suffer from slow FPU (Floating Point Unit) operation, which is crucial to the Mellosotron's performance.



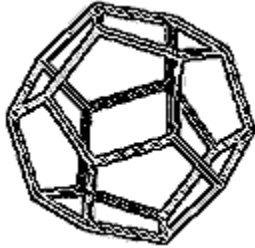
Tommy Anderberg is the creator of several popular audio tools for MS Windows ([Acid WAV](#), [MIDnight Express](#), [Virtual Sampler SDK](#), [WAVmaker](#)). A physicist by training and a computer geek by vocation, he is also a consultant and a prolific freelance writer on personal computing and the Internet.

Tommy can be reached at

[Tommy.Anderberg@polyhedric.com](mailto:Tommy.Anderberg@polyhedric.com)

## Other titles from Polyhedric Software

For the latest information on titles published by Polyhedric Software, please visit our web site at



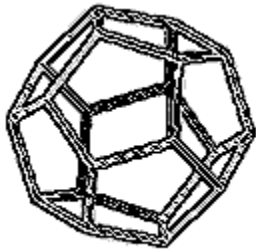
[www.polyhedric.com/software](http://www.polyhedric.com/software)

Current titles of interest to users of the Mellosoftron include:

- The Acid WAV sound editor and synthesizer. The best tool for recording, synthesizing, editing and compressing WAVs!
- The WAVmaker CD+ quality MIDI to WAV renderer featuring programmable effect lines, tune tables, the GSound 44 instrument library and much more!
- The MIDnight Express wavetable-based MIDI player and MIDI to WAV renderer. An ideal companion for the Mellosoftron - and FREEWARE!
- The GSound 22 CD-ROM: a complete General MIDI and GS sound library for the Mellosoftron and the MIDnight Express.
- The Virtual Sampler SDK. A Software Developer's Kit which lets programmers use the Mellosoftron's low latency wavetable engine in their own projects.

We also undertake custom software design and development on a contract basis. You are always welcome to contact us at [consulting@polyhedric.com](mailto:consulting@polyhedric.com) to discuss your custom development needs.





Polyhedric Software

Acid WAV is an advanced sound editor and synthesizer. Unlike older (and more expensive) sound editors, it has been designed from the ground up to take full advantage of Microsoft's 32 bit Windows architecture.

An unparalleled set of powerful synthesis and editing functions combined with an intuitive, friendly user interface make Acid WAV a must-try (and a must have!) - words just won't make it justice! Grab the free Acid WAV Evaluation Package at our site

[www.polyhedric.com/software](http://www.polyhedric.com/software)

and see for yourself!

The screen shots below show just a few of Acid WAV's windows. Download the Evaluation Package to explore them all!

[Main window](#)  
[Convolution filter](#)  
[Virtual room](#)  
[Frequency envelope](#)  
[FM synth](#)  
[Analog synth](#)  
[Equalizer](#)  
[Spectrum](#)  
[Script editor](#)

Acid WAV is only \$30 when ordered with the Mellosotron as part of the [WAVmaker Studio package!](#)

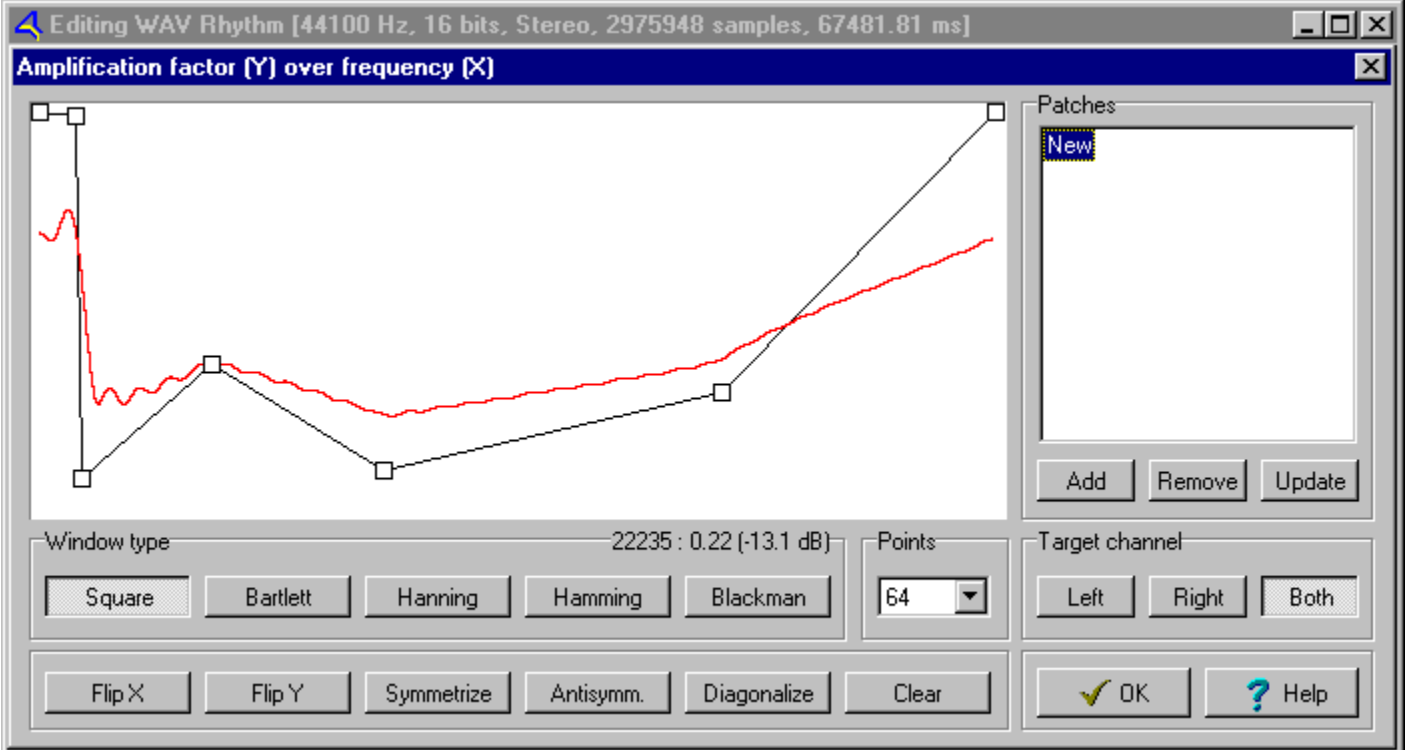
[Other titles from Polyhedric Software...](#)

Editing WAV Rhythm\* [44100 Hz, 16 bits, Stereo, 78309 samples, 1775.71 ms] 57377 : 6907

Basic | Time | Amplitude | Frequency | Synthesis

Trim  
Cut  
Copy  
Delete  
Paste  
Paste mix  
Insert file  
Mix file(s)  
Insert silence  
Swap channels  
Duplicate channel  
Convert to stereo  
Convert to 8 bits

Start 11563 End 21223  
Horizontal zoom 1.00 Vertical zoom 3.00  
Reset Milliseconds Samples Help



Basic Time

### Virtual room

- Ech
- Hall rev
- Chapel r
- Chamber
- Custom r
- Virtual r
- Rever
- Stretc
- Tremc
- Vibra
- Chor
- Flang
- Ring mod

Room size (m)

Height (Z)

Length (Y)

Width (X)

Air absorption (dB/m)

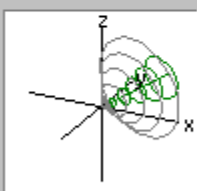
0.1

Reflectivity

North (0°)	<input type="text" value="70%"/>
East (90°)	<input type="text" value="70%"/>
South (180°)	<input type="text" value="70%"/>
West (270°)	<input type="text" value="70%"/>
Ceiling (+90°)	<input type="text" value="55%"/>
Floor (-90°)	<input type="text" value="25%"/>

Left speaker | Right speaker | Left microphone | Right microphone

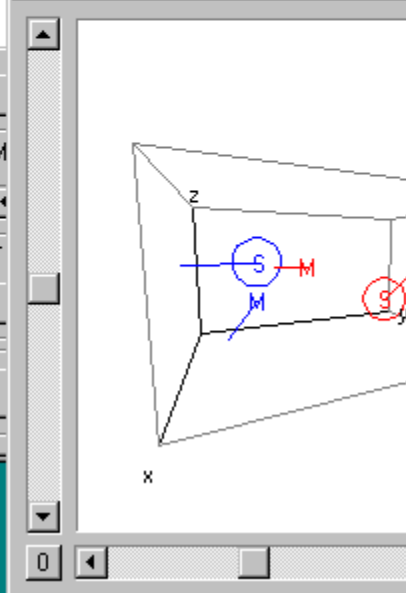
Z (m)	<input type="text" value="1.00"/>	Elevation (°)	<input type="text" value="35.0"/>
Y (m)	<input type="text" value="8.00"/>	Direction (°)	<input type="text" value="95.0"/>
X (m)	<input type="text" value="1.00"/>	Inner cone (°)	<input type="text" value="30.0"/>
Size (m)	<input type="text" value="1.00"/>	Outer cone (°)	<input type="text" value="90.0"/>

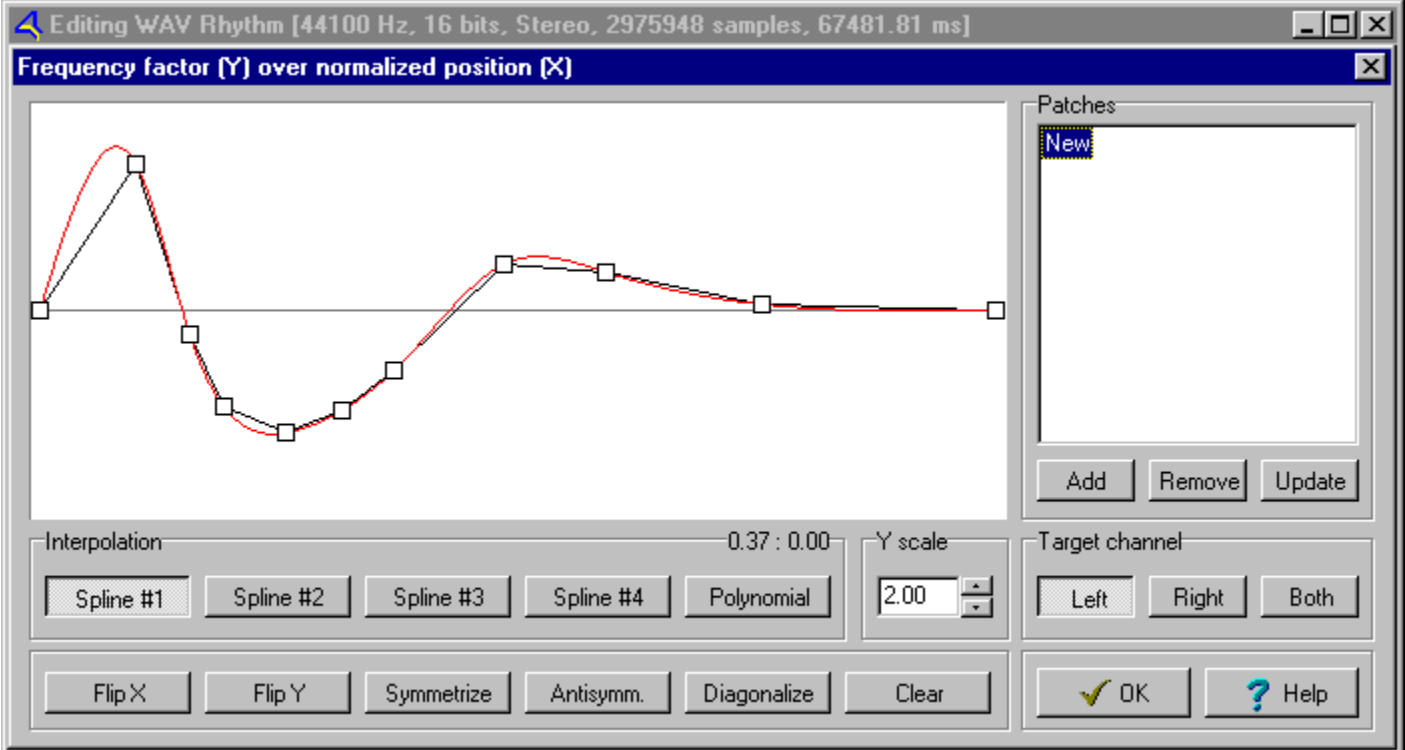
 Close room display

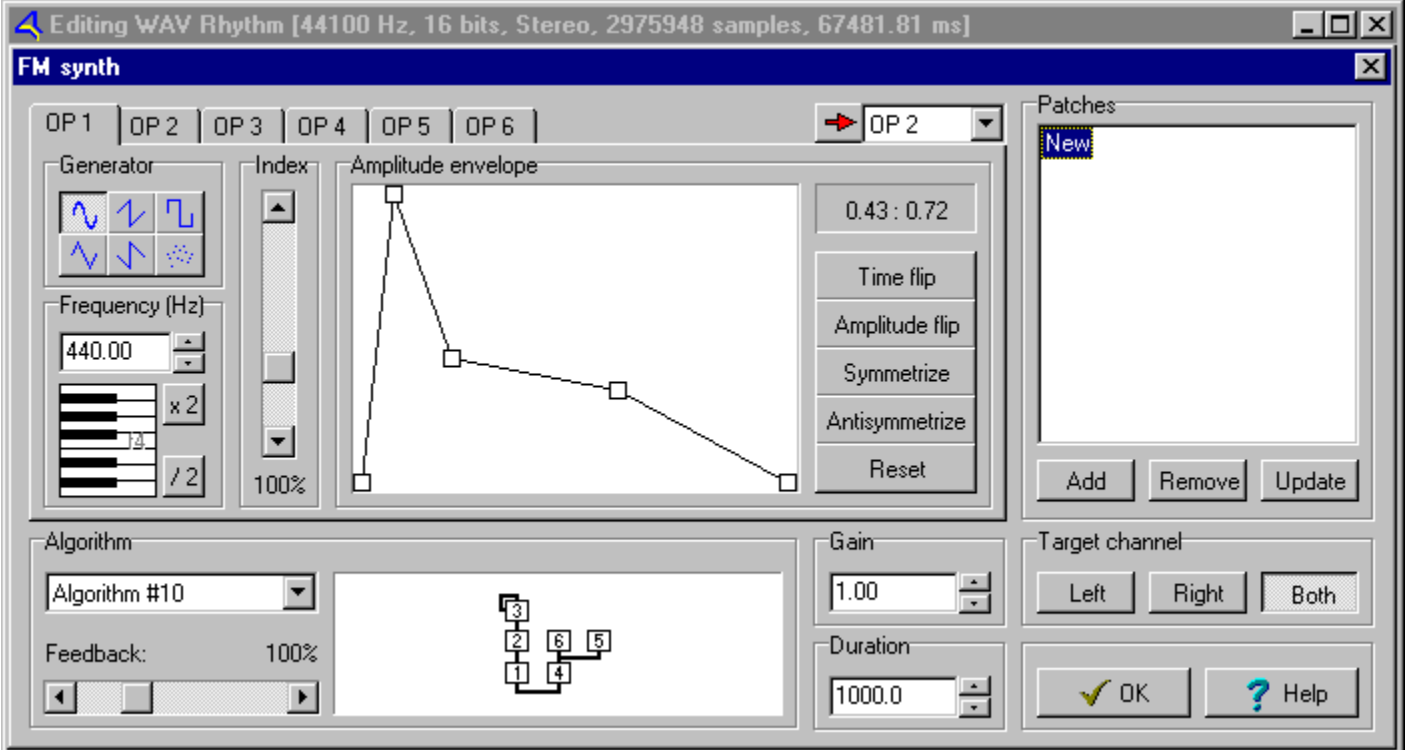
Patches

New

### (S)peaker and (M)icrophone positions







Editing WAV Rhythm [44100 Hz, 16 bits, Stereo, 2975948 samples, 67481.81 ms]

### Analog synth

OSC 1 | OSC 2      Sync lock OSC 2 to OSC 1

**Generator**

Filter

Center (Hz) 5000

Resonance (%) 0

Frequency (Hz) 436.00

**Normalized phase over time (one cycle)**

0.78 : 0.98

Time flip

Phase flip

Symmetrize

Antisymmetrize

Reset

**Patches**

New

Add Remove Update

**Target channel**

Left Right Both

Copy OSC 1 → 2

Mix 78% OSC 1 | 22% OSC 2

Gain 0.90

Duration 1025.0

OK Help

Editing WAV Rhythm [44100 Hz, 16 bits, Stereo, 2975948 samples, 67481.81 ms]

### Equalize

**Left**

+ 30 dB  
± 0  
- 30 dB

-1 -2 -4 -8 -12 -16 -20 -30 -40 -50 kHz

0 - 1 - 2 - 4 - 8 - 12 - 16 - 20 - 30 - 40 - kHz

**Right**

+ 30 dB  
± 0  
- 30 dB

-1 -2 -4 -8 -12 -16 -20 -30 -40 -50 kHz

0 - 1 - 2 - 4 - 8 - 12 - 16 - 20 - 30 - 40 - kHz

Patches

New

Add Remove Update

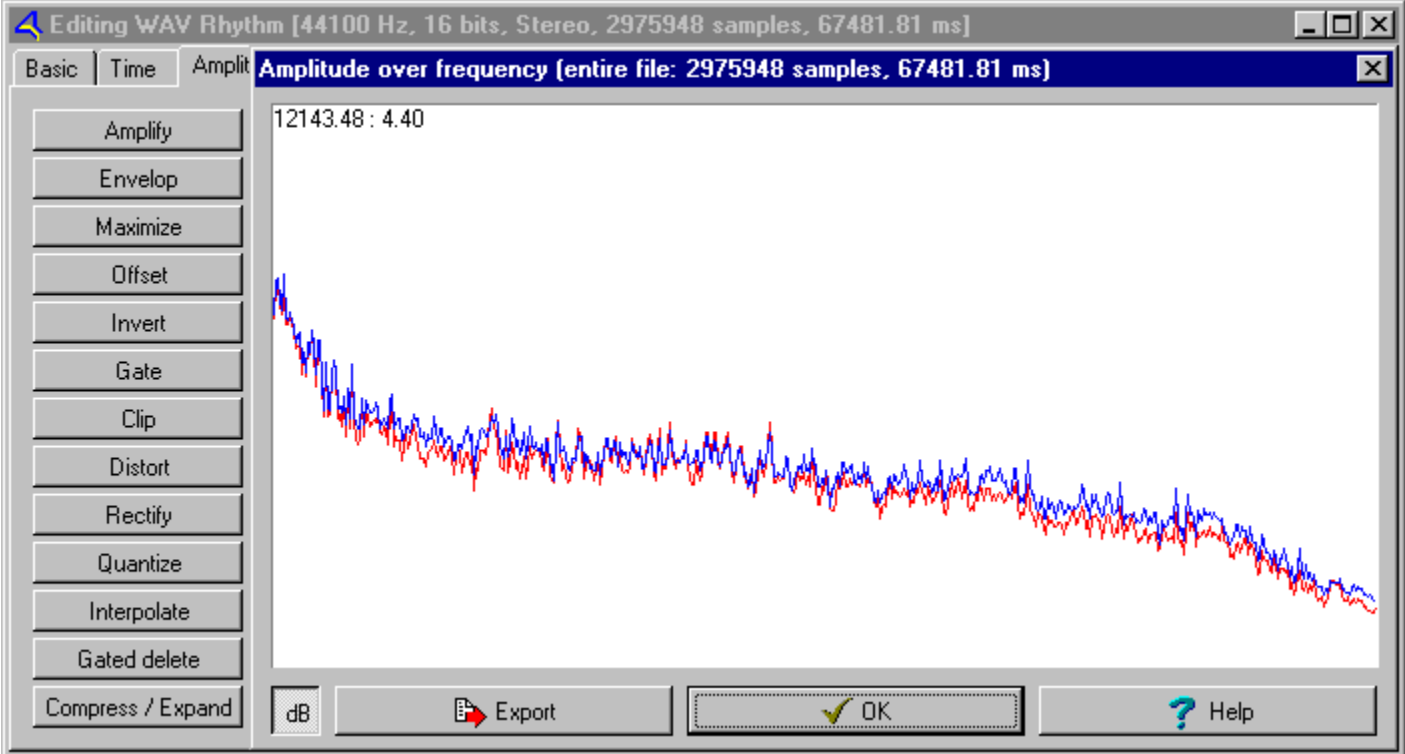
Target channel

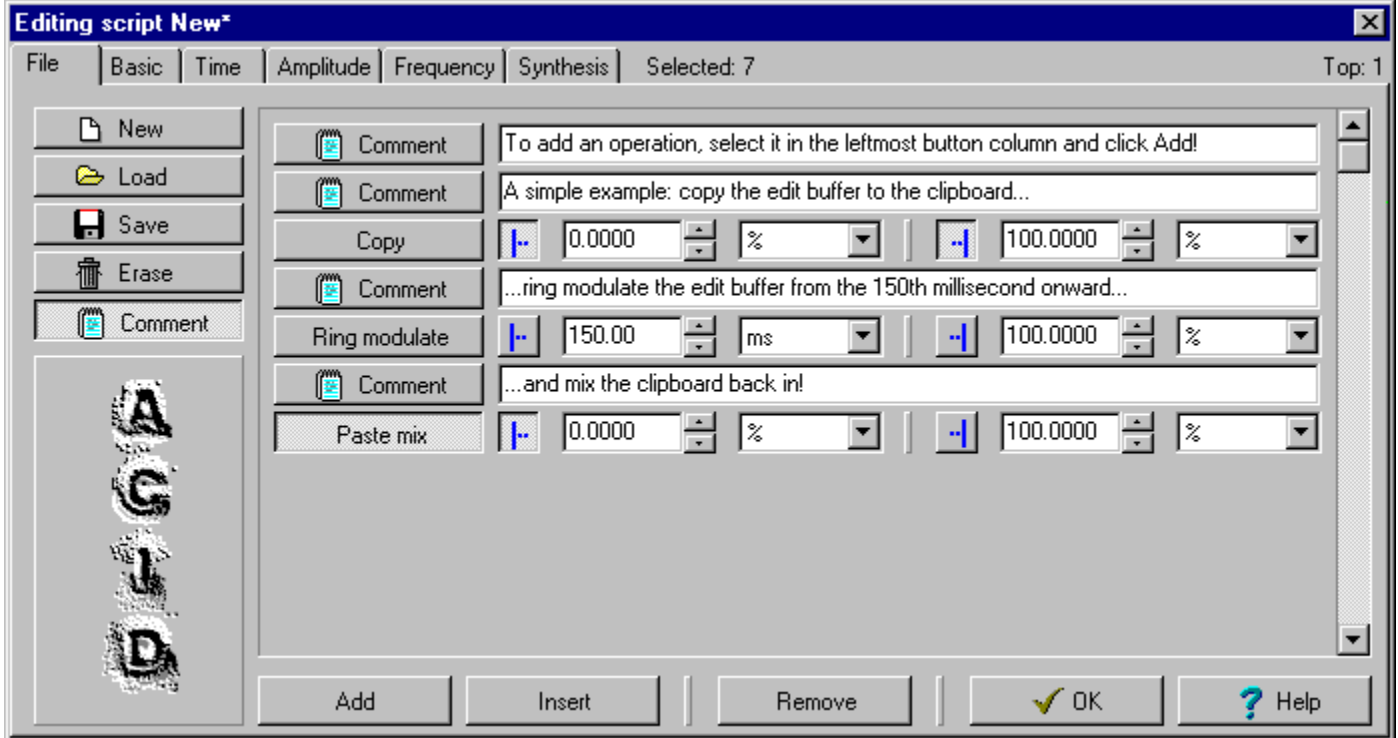
Left Right Both

FFT OK Help

The image shows a software interface for equalizing audio. The main window title is "Editing WAV Rhythm [44100 Hz, 16 bits, Stereo, 2975948 samples, 67481.81 ms]". The dialog box is titled "Equalize". It features two sections for "Left" and "Right" channels. Each section has ten vertical sliders representing frequency bands, with labels below them: -1, -2, -4, -8, -12, -16, -20, -30, -40, and -50 kHz. The top of each slider is marked with a horizontal line, and the right side has a scale from +30 dB to -30 dB, with a "± 0" button. Below the sliders, there are two rows of frequency markers: the top row has -1, -2, -4, -8, -12, -16, -20, -30, -40, -50 kHz, and the bottom row has 0, 1, 2, 4, 8, 12, 16, 20, 30, 40 kHz. To the right of the sliders is a "Patches" list containing a single entry "New". Below the list are "Add", "Remove", and "Update" buttons. Underneath is a "Target channel" section with "Left", "Right", and "Both" buttons. At the bottom right are "FFT", "OK", and "Help" buttons.







## About the MIDNIGHT Express

The MIDNIGHT Express is a free, programmable, software wavetable MIDI player and MIDI to WAV renderer which uses the same instrument file (PRG and WAV) format used by the Mellosotron.

Unlike the Mellosotron, the MIDNIGHT Express doesn't require (but nevertheless supports) DirectSound and a fast CPU. In fact, it will even run on a low end 486 PC. Any entry-level 16 bit sound card, or even the simple sound circuits found in most laptops, will do just fine.

You can use the MIDNIGHT Express...

- [As a standalone MIDI player](#)
- [As a fast MIDI to WAV renderer](#)
- [As a better WAV player](#)

And it's easy, too! If you have ever used a CD player, you will feel right at home with the MIDNIGHT Express II's user interface:



If you need CD+ quality MIDI to WAV rendering with advanced features like programmable effects and tune tables, check out [WAVmaker!](#)

[Other titles from Polyhedric Software...](#)

If you are tired of listening to your MIDI files as rendered by the toy FM synth chips found on Sound Blaster-compatible sound cards and in most laptops, the MIDInight Express is for you! Pop the [GSound 22 CD-ROM](#) into your PC, set the MIDInight Express playing, and enjoy!

Sometimes you need to create a song on WAV format for a web page, a program, or a multimedia application.

The MIDInight Express offers the fastest route - just click the record button and play your MIDI file(s) as usual! In most cases, the result will sound much better than recording off your sound card while playing.

This function is also useful for listening to MIDI files which are too complicated for real time rendering on your current hardware (too little free RAM to hold all samples, causing the disk swapper to kick in, or too slow a CPU). Note that the MIDInight Express can play WAVs, too.

A tip for all webmasters: Acid WAV is not only the perfect tool for recording, synthesizing and editing sound; it's also great for compressing WAVs created with the MIDInight Express before posting them on the web, e.g. in MPEG format.

If you need CD+ quality MIDI to WAV rendering with advanced features like programmable effects and tune tables, check out WAVmaker!

An auxiliary function of the MIDInight Express is the ability to play WAV files using the standard Media Control Interface (MCI). On most systems, it will do a better job than the Windows 95 media player (which tries to load the whole WAV before playing it, bringing the whole system to a virtual standstill when there isn't enough free memory).



WAVmaker is the undisputed leader in MIDI to WAV rendering. Unlike sound cards and soft synths like the Mellosoftron and the MIDnight Express, which are constrained by the requirements of real time operation within a limited memory and CPU footprint, WAVmaker turns MIDI files of virtually unlimited polyphony and complexity into CD-quality WAVs with perfect timing precision, using fully editable instrument patches and samples of virtually unlimited size, without the noise and distortion introduced by conventional recording.

The WAVs produced are ready to be burnt on CD, uploaded to a website or processed further (e.g. mixed with recorded vocals).

Version III is a quantum leap in WAVmaker's development. The program has been completely rewritten to take full advantage of current state of the art PC multimedia technology. New features include:

- New, higher precision rendering kernel
- Programmable, MIDI-controlled effect chains
- User-definable tune tables
- Automatic volume optimization
- Batch rendering
- Wider target sampling rate range
- Improved MIDI reporting and editing
- Integration with the Acid WAV sound editor and synthesizer
- Integration with the Mellosoftron
- New, Explorer-style user interface

Last but not least, WAVmaker now comes with the **GSound 44** General MIDI and GS instrument library: a CD-ROM with full, ready-to-use, MIDnight Express- and Mellosoftron-compatible GM and GS sets consisting of 3+ MB PRG patches and 261+ MB 44.1 kHz, 16 bit WAV samples!

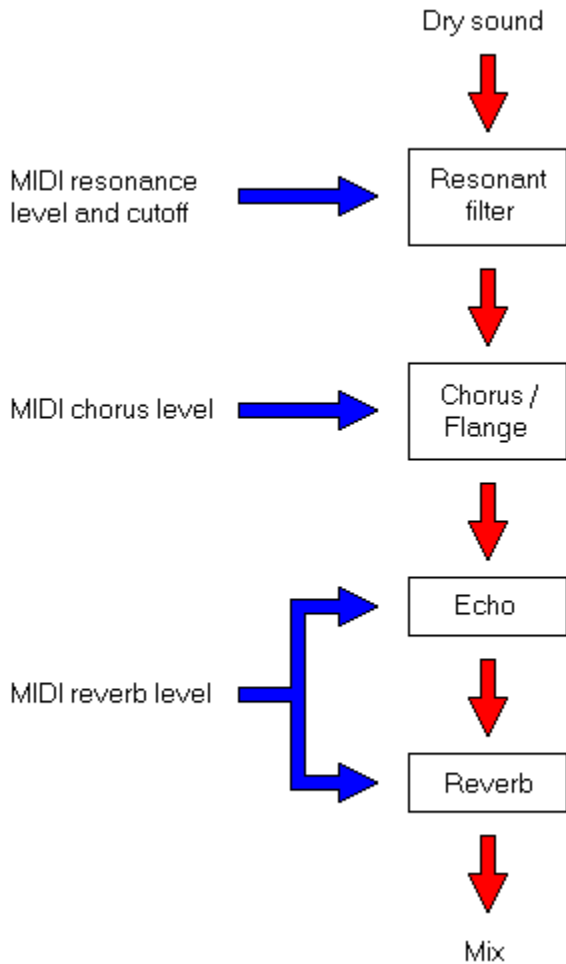
The Mellosoftron is only \$20 when ordered as part of the WAVmaker Plus or WAVmaker Studio packages!

Other titles from Polyhedric Software...



Advances in CPU performance have finally made the use of floating point arithmetic for the entire rendering process a practical proposition. All internal operations are now performed in 32 bit precision or higher, for a signal to noise ratio of no less than 192.6 dB. The results are reduced to (and optimized for) standard 16 bit (CD / DAT) sample size in a second pass, after the final mixdown has been created.

Each MIDI track and channel can be assigned its own MIDI-controlled effect chain with the following architecture:



The chorus / flange, echo and reverb units are all programmable.

This means that you can have a different effect chain for each track and channel. Hardware sound modules typically give you a single, shared effect chain and only let you vary the wet mix level individually for each channel.

Chorus / flange, echo and reverb patches are easily created, edited and tested using the dedicated Acid WAV editor and synthesizer (included!). Import / export of Acid WAV effect patches is done at the click of a button.

All effects can be turned off if you prefer a dry mix (e.g. when rendering individual tracks for manual processing).

Tired of that old Western-style tuning? Feel like experimenting with microtonal composition? No problem: each MIDI track and channel can be assigned its own individual tune table. WAVmaker comes with several predefined ones and gives you full liberty to create your own tunings.

The two-pass approach used by the new rendering kernel means that overall volume no longer needs to be adjusted manually: the resulting WAVs are automatically optimized for the target 16 bit dynamic range.

Using WAVmaker has never been easier: all you have to do is collect the MIDIs and RMIs to render in a list, click the magic button and watch WAVmaker launch a separate rendering process. You are then free to terminate the main program. This minimizes WAVmaker's footprint while rendering. You can also collect a new bunch of MIDI files to render with different settings and launch another job while the first one is still running. There is no limit to the number of jobs you can have running in parallel (other than available memory etc.). This feature is particularly useful if you have multiple CPUs.

Supported target sampling rates range semi-continuously (i.e. in increments of 1 Hz) from 8000 Hz (phone quality, used e.g. in Unix AU files) to 48 kHz (DAT quality).

WAVmaker now comes with a dedicated MIDI reporting and editing "wizard". The new tool analyzes the contents of a MIDI file and lets you edit its individual tracks and channels. You can delete selected kinds of events; map GS banks, programs and controllers to each other; scale and offset continuous values; and transpose note ranges. The result can be saved to any of the supported MIDI formats: Standard MIDI Level 0, Standard MIDI Level 1, RIFF MIDI Level 0 and RIFF MIDI Level 1.

Acid WAV doesn't only serve as WAVmaker's effect patch editor: it also replaces the auxiliary sound editing functions found in WAVmaker 1.x and 2.x.

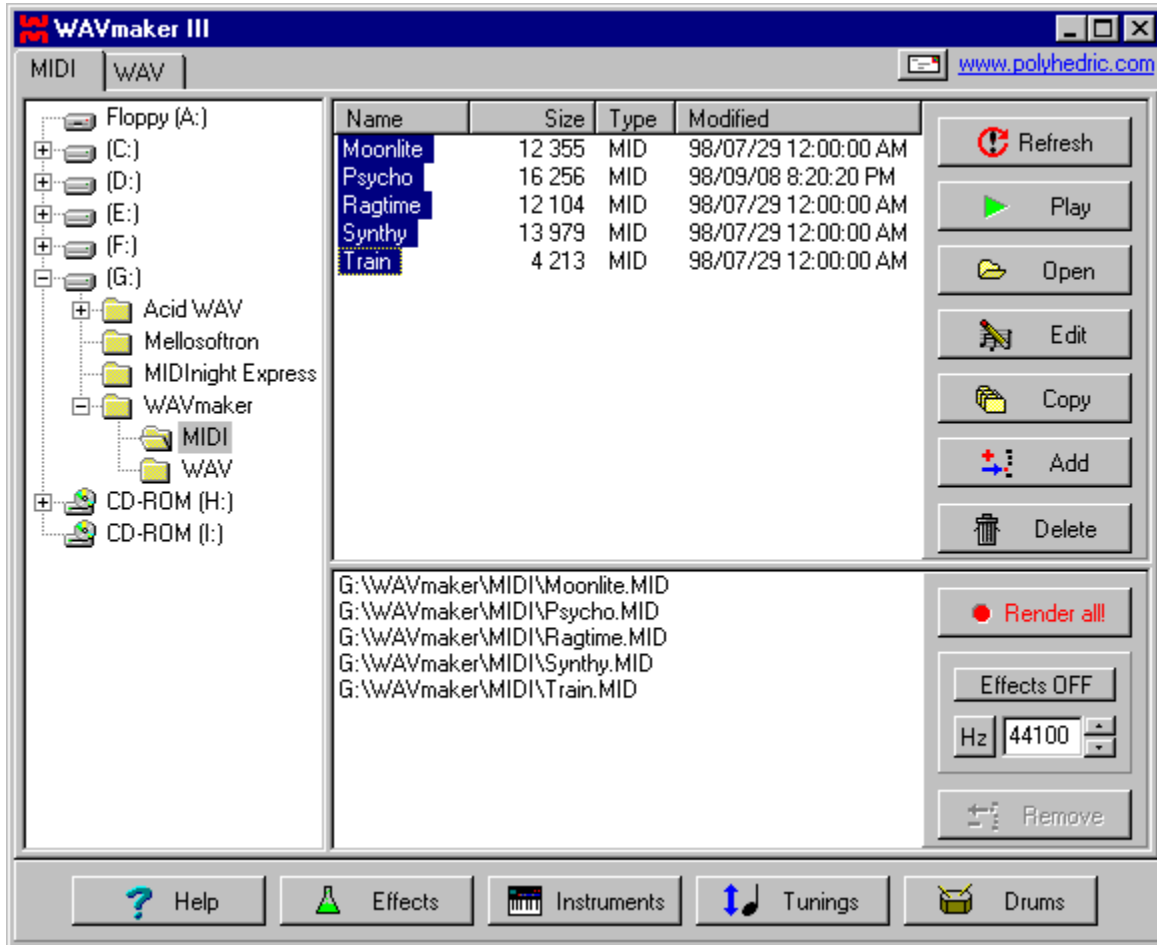
That's right, you now get Acid WAV with WAVmaker!

To make a good story even better, you can choose the level of bundling that suits you best. WAVmaker's Standard license covers Acid WAV's recording, encoding, decoding and basic editing functions. Advanced users can take advantage of the Studio license to get all of Acid WAV's features.



Since WAVmaker now fully supports Mellosotron 2.x patches, there is no longer any need for a built-in patch editor. Using the Mellosotron makes life a lot easier by allowing patches to be modified and tried out immediately, live at your keyboard. Again, you can choose the level of bundling that suits you best: if you are not interested in creating your own instruments, it's enough to get a Standard license; otherwise, you can choose between the WAVmaker Plus license (which includes the Mellosotron) and the WAVmaker Studio license (which adds Acid WAV's advanced sound editing and synthesis functions).

WAVmaker's user interface has been completely redesigned to present you with a greatly simplified, familiar environment. Windows integration has also been improved with the addition of drag & drop (both internally and from the Windows Explorer) and of Open / Play buttons which rely on your system settings to launch the appropriate viewer / player (e.g. a MIDI sequencer or the Windows Media Player).



Other titles from Polyhedric Software...



The GSound 22 CD-ROM is only \$29...  
...or \$20 if ordered together with the Mellosotron!

We burn your GSound 22 CD-R when we receive your order. Keeping no stock allows us to minimize costs and guarantees that you always get the most recent version of all files.

- To submit your order online, you need a credit card. A secure form is available for browsers supporting SSL (e.g. newer versions of Netscape Navigator and Internet Explorer). Data entered on the secure form is encrypted before it's sent over the Internet.



[Click here to submit your order online.](#)

- To submit your order by e-mail, fax or snail mail, you need to run Kagi's "Register" wizard. It will prompt you for all the necessary information and encode any credit card data for better security.



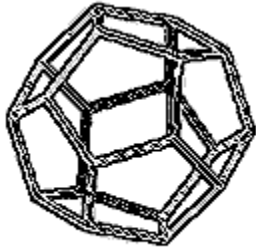
[Click here to submit your order by e-mail, fax or snail mail.](#)

If Windows Help complains about not being able to find Register.EXE, you can launch it from the Mellosotron's folder (click the icon labeled "Order").

Please direct any questions about ordering our products to [sales@polyhedric.com](mailto:sales@polyhedric.com).

# Virtual Sampler SDK

Copyright © Tommy Anderberg, 1997-98



Polyhedric Software

The Mellosoftron is a GUI built on top of the **Virtual Sampler Library** (TM). All the MIDI to digital audio functionality - from loading PRGs and WAVs to interpreting and acting on MIDI messages - is provided by the Virtual Sampler Library.

As a developer using the Virtual Sampler Library, you deal with a simple API emulating a real sampler. You control it using standard MIDI messages, with one difference - there are 65536 channels to choose from, not 16! The difference between MIDI and digital audio ceases to exist, and you can use the same tool for everything from music - with predicatable results, independent of the end user's MIDI hardware - to sound effects. The Virtual Sampler Library coexists happily with straight DirectSound code, too (but chances are that you will not see any reason to deal with the latter).

The full Virtual Sampler Library API counts only 27 functions, most of them used to (optionally) set parameters such as the latency and the current WAV directory. Only a third of them are required to build a full application. This simplicity, the use of the well-known MIDI standard, and the availability of powerful PRG and WAV tools like the Mellosoftron and Acid WAV all make using the Virtual Sampler Library a breeze.

The **Virtual Sampler SDK** contains the documentation and code needed to use the Virtual Sampler Library in your own programs. It also includes the Mellosoftron for quick sound patch creation and editing.

The following 32 bit development environments are currently supported by the Virtual Sampler SDK: Watcom C/C++, MS Visual C/C++, Borland C++ Builder, Borland Delphi, MS Visual Basic.

A fully functional demo is available for download from our site,

[www.polyhedric.com/software](http://www.polyhedric.com/software)

Other titles from Polyhedric Software...

## Playing the Mellosotron live

Before you can do any playing, you must

- learn about instrument pads
- load a PRG (instrument patch file) on at least one pad.

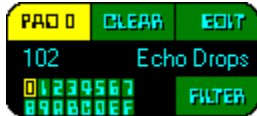
Once you've done so, there are three ways to actually play the Mellosotron live:

- With a MIDI keyboard (or other MIDI controller) hooked up to your PC's MIDI In port.
- With the mouse.
- With the computer's alphanumeric keyboard.

One method doesn't exclude the other - you can use all three at the same time.

## About instrument pads

The Mellosofron's main window is dominated by 16 rounded rectangles arranged in four rows and labeled 0 through F (you may recognize this as hexadecimal notation for the numbers 0 through 15). They are referred to as (instrument) **pads**:



Each pad is essentially a separate, polyphonic, MIDI-controlled instrument.

To see which instrument patch (**PRG**) is assigned to a pad, look at the black area at the pad's center. The PRG's filename is displayed first (in the picture, it's "102") and there may also be a clear-text description following it (here, it's "Echo Drops"). Clicking the black area causes note c4 to be played.

To edit the PRG, click the **Edit** button (PRG editing is covered in detail [here](#)). To unload the PRG, click the **Clear** button.

A pad with no PRG assigned to it shows a slashed line where the PRG name and description would otherwise be displayed:



Clicking the black area of an unassigned pad has no effect.

To load a PRG, click **Load** (loading PRGs is covered in detail [here](#)). To create a new PRG from scratch, click **New** (creating PRGs is covered in detail [here](#)).

## MIDI settings

Each pad can read incoming MIDI messages on any and all active MIDI channels, i.e. on any MIDI channel that's been highlighted in the global MIDI channel selector, immediately below the pads:



Use the channel selector below the pad's PRG line to edit the pad's MIDI settings. In the pictures above, the pad is set to respond to messages on MIDI channel 0. In the picture below, the pad is set to respond to messages on MIDI channel 1, 5 and 14 (hexadecimal E):



**IMPORTANT: the pad's MIDI settings are overruled if Omni is on** (see below).

## About the selected pad

In all pad pictures above, the pad's upper left corner is highlighted (black text on yellow background). This means that the pad is [selected](#). All notes played on the Mellosotron's graphic keyboard and on the computer's alphanumeric keyboard, as well as all controller events originating from the mousepads above the graphic keyboard, are routed to the selected instrument pad.

Additionally, if **Omni** is on (i.e. if the Omni button immediately below the pads is highlighted),



all incoming MIDI messages on all active channels are rerouted to the selected pad, irrespective of the pad's MIDI settings. This is equivalent to disabling all other pads while setting the selected pad to read all MIDI channels (but obviously much quicker!). It's particularly convenient when controlling the Mellosotron from a MIDI keyboard: instead of sending MIDI Program Change messages, load the instruments you'll be needing, set Omni on, then quickly switch between instruments by selecting the appropriate pad.

To select a pad, click its upper left corner. You can also step through the pads using the arrow buttons next to the graphic keyboard:





## Loading an instrument patch (PRG)

You load individual PRGs by assigning them to pads (you can also load entire instrument groups using MAPs).

To start with, locate a pad with no PRG assigned to it. The central (black) part of such a pad is filled with a slashed line:



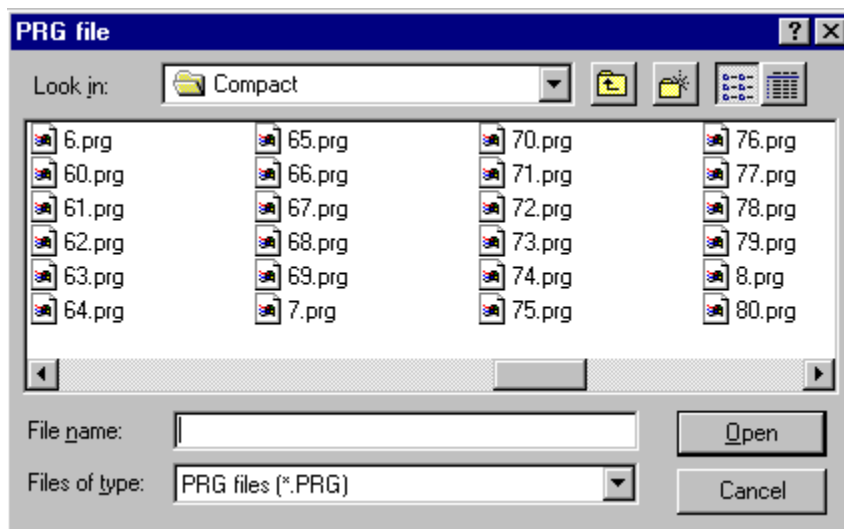
A pad with a PRG assigned to it on the other hand looks more like this:



The PRG's filename is displayed, and there may also be a clear-text instrument description following it. If you click the black area, you will hear note c4 being played with the currently loaded PRG.

If all pads are taken, just pick one and click its **Clear** button. This causes the PRG on that pad to be unloaded and the Clear button to turn into a **Load** button.

Click the Load button! You will be presented with a standard Open File dialog giving you a view of the current PRG directory:



Select a PRG file and click **Open**! If all goes well, the instrument patch will be loaded and the pad will be activated.

If the pad is not activated, there was something wrong with the patch - start by checking that the current WAV directory contains the WAVs referenced in the PRG which you tried to load. PRGs are just formatted text files, so you can browse them with any text editor, like Windows Notepad.

You can also drag a PRG file e.g. from Windows Explorer and drop it on the main window. It will be assigned to the first free pad.

Once a pad has been activated, it starts responding to incoming MIDI messages, including Program Change events. If you want to control PRG loading through MIDI, it's convenient to set up the default MAP to load an instrument on all pads at program startup.

MIDI Program Change events are interpreted according to the same rules used by the MIDInight Express and WAVmaker: a change to MIDI program number 0 causes 0.PRG to be loaded, unless it occurs on [drum channel](#), in which case D0.PRG is loaded.

Bank Select (Controller 0) messages, i.e. GS variations, are also supported. A filename on the form 78b16.PRG denotes instrument number 78, bank number 16. When a GS variation is requested, the Mellosotron tries to find the corresponding PRG file; if that fails, the corresponding default (bank 0) PRG is loaded. For instance, failing to find 78b16.PRG would cause 78.PRG to be loaded instead.

Note that Bank Select messages are not acted upon until a Program Change message is received, and remain in force until the channel is reset or a new Bank Select message is received.

## About PRGs

The Mellosoftron needs two kinds of files to fully describe an instrument: PRGs and WAVs.

PRGs do not contain any sound data. The actual samples are kept in WAV files. PRGs assign WAV files to keys (notes) and impose performance parameters (base volume and pan position, controller sensitivities, envelope shape, loop points...). Each key on the 128-note MIDI keyboard can have its own individual sample file(s) and performance parameters.

While you can use any text editor to create and modify PRG files, the recommended tool for this task is the Mellosoftron's dedicated PRG editor.

The Mellosoftron wants all the PRGs needed to handle MIDI Program Change events to reside in a common directory, so it's a good idea to organize your PRGs in "libraries", with each library corresponding to a separate directory.

Click the **PRGs** button in the main window to change the current PRG directory.

## About PRG filenames

The Mellosoftron allows you to give a PRG any valid filename, not just one adhering to the MIDInight Express and WAVmaker convention based on MIDI program and bank numbers. If you want to create a "Killer Bass 2.PRG", you can - but you won't be able to use it with WAVmaker and with the MIDInight Express.

There is a much better way to associate PRGs with clear-text instrument descriptions. In the Mellosoftron's directory, locate a file called ML.INS, copy it to the PRG directory and open the copy with any text editor (e.g. Windows Notepad). You will see a bunch of lines similar to this one:

```
38b1=GS Synth Bass 101
```

Each line contains a PRG name (no directory, no extension, just the name; here it's "38b1") and the corresponding clear-text instrument description, separated by the "=" character (any line which doesn't contain this character is considered a comment and simply ignored by the Mellosoftron). When a PRG is loaded, the Mellosoftron looks up the corresponding description and displays it on the instrument pad alongside the PRG name.

You can add your own lines and edit existing ones freely. The file ML.INS is loaded when you change PRG directory, so you need to reselect the directory when you're done editing the file and have saved it back to disk.

If the current PRG directory doesn't contain an ML.INS file, the Mellosoftron will use the ML.INS file in its own directory. While you can edit this file too, doing so is not recommended. It's generally better to leave the GM/GS standard instrument descriptions in this file alone and override them with an ML.INS in the target PRG directory. This way, you can give all PRG libraries you create their own descriptions, without having to worry about potential name conflicts.

## About WAV paths

Normally, the Mellosoftron expects to find the WAVs referenced by PRGs in the current WAV

directory (selected by clicking the **WAVs** button in the main window). If you try to load a PRG and nothing seems to happen, the most likely cause is that the PRG references a WAV which can't be found in the current WAV directory.

As of version 2.1, the Mellosotron also allows you to use full WAV paths in PRGs, overriding the current WAV directory. This feature has been added upon the request of some users, but its use is not recommended. A PRG containing a full WAV path is totally dependent on your system's directory structure to function properly. It will most probably not work on other machines (so you can't exchange such patches with other users) and will "break" on your own system as soon as you start moving WAV files or directories around.

## About WAVs

The Mellosoftron needs two kinds of files to fully describe an instrument: PRGs and WAVs.

WAV files contain the actual sound data (samples) referenced by PRG instrument definitions. Like the MIDInight Express, the Mellosoftron will handle any combination of mono and stereo, 8 or 16 bits/sample and sampling rates other than the standard 11025, 22050 and 44100 Hz. Note however that WAVmaker requires sample rates of 44100 Hz or higher.

Only PCM (unencoded) WAVs can be used. Encoded WAVs can be decoded for use with the Mellosoftron using Acid WAV.

Up to and including version 2.0, all WAVs referenced by a PRG had to reside in a common directory. This is still the recommended mode of operation. Click the **WAVs** button in the main window to change the current WAV directory.

As of version 2.1, the Mellosoftron also allows you to use full WAV paths in PRGs. This feature has been added upon the request of some users, but its use is not recommended. A PRG containing a full WAV path is totally dependent on your system's directory structure to function properly. It will most probably not work on other machines (so you can't exchange such patches with other users) and will "break" on your own system as soon as you start moving WAV files or directories around.

## Playing with a MIDI keyboard

There is not much to say about using a MIDI keyboard. It's just like playing an ordinary sampler. Apart from Note On/Note Off events, the Mellosofron implements all standard controller (Volume, Pitch Bend, Modulation, Pan Position, Pressure, Expression, Bank Select) and Program Change events.

In addition, the Mellosofron implements the following non-standard controller events (also used by WAVmaker and Yamaha XG devices):

- 47H: Resonant filter level
- 4AH: Resonant filter cutoff

The filter on the target pad must be enabled for these events to have any effect:



Note that any instrument pad can read any combination of MIDI channels - pad 0 need not be set to channel 0, for instance. If you have mapped a MIDI channel to an instrument pad with no PRG assigned to it, don't be surprised if you can't hear a thing when playing on that channel! Same goes if you have selected a different MIDI input port than the one to which you have hooked up your keyboard.

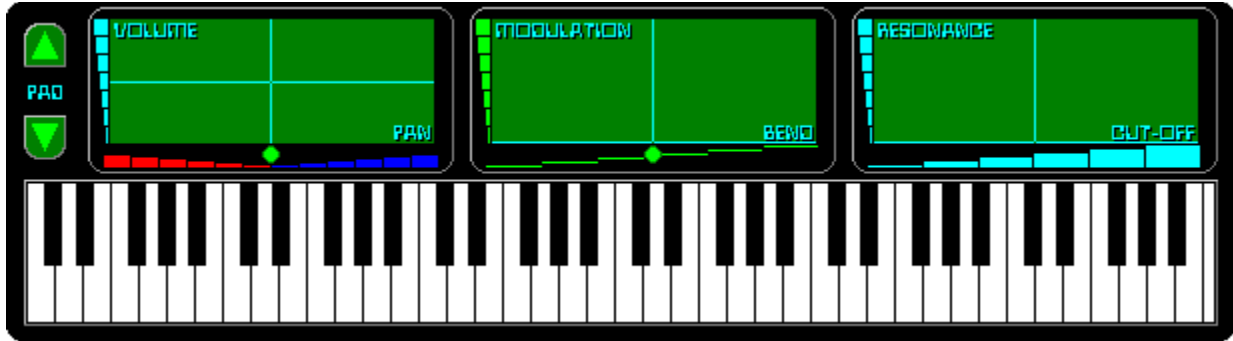
A more serious, potential source of trouble is that some sound cards and/or drivers can't handle MIDI input and digital audio output at the same time. On such a system, the Mellosofron will report that MIDI input and/or digital audio output aren't working already at startup, and give you the option to abort execution then and there.

Finally, remember that when **Omni** is on, all incoming MIDI messages are redirected to the selected instrument pad, independently of the channel on which they originated. See the section on instrument pads for details on Omni mode.

## Playing with the mouse

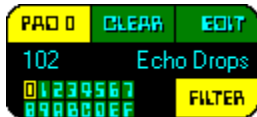
The Mellosoftron can be played by clicking the graphic keyboard displayed at the bottom of the main window. Sustained instruments will keep playing until you release the mouse button.

You also have three "mousepad" controls at your disposal:



Each mousepad is linked to a MIDI controller pair. Click and/or drag the mouse cursor on the central (green) area to modify both controller values. Use the left and bottom axis areas to modify only one controller value.

All events generated by the graphic keyboard and mousepads are sent to the currently selected instrument pad. Note that the selected pad's filter must be enabled for Resonance and Cut-off events to have any effect:



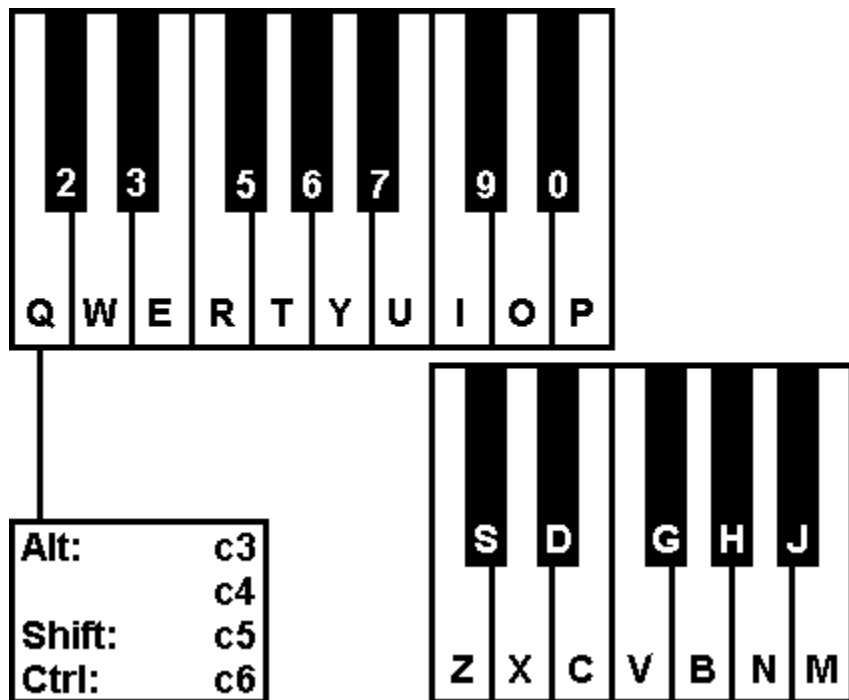


## Playing with the computer keyboard

When the main window has input focus, you can play the Mellosotron using the computer's alphanumeric keyboard.

All events generated this way are sent to the currently selected instrument pad. Attack and release velocities are always set to 64.

As of version 3.0, the mapping from alphanumeric keys to MIDI notes is user-definable. The default map looks like this:



German-language users with QWERTZ-type keyboards and others who wish to change the notes assigned to each key can easily do so by editing the file ML.KCT in the Mellosotron's directory using any text editor (e.g. Windows Notepad).

The file format is quite simple: each data line consists of a symbolic virtual key code (don't worry if you don't have a list of virtual key codes, they are all in the default ML.KCT file) and a MIDI note number (or offset), separated by the "=" character. For instance, the line

```
VK_Q=48
```

assigns MIDI note number 48 to the Q key. This assignment can be modified by the three modifier keys (Alt, Shift and Ctrl). In the default ML.KCT file, these are assigned as follows:

```
# ALT key  
VK_MENU=-12
```

```
# SHIFT key  
VK_SHIFT=12
```

```
# CTRL key  
VK_CONTROL=24
```

With these assignments, pressing Alt-Q results in MIDI note  $-12 + 48 = 36$  being played, Shift-Q yields  $12 + 48 = 60$  and Ctrl-Q yields MIDI note  $24 + 48 = 72$ .

Note the lines starting with "#" characters in the example above. The "#" is just an aid for readability; any line which doesn't contain the "=" character is considered a comment and simply ignored by the Mellosotron.

You should also be aware of the special "note number" -1. This translates to an All Notes Off event. The default ML.KCT file contains the line

```
VK_SPACE=-1
```

so with this mapping, pressing the spacebar causes all notes on the selected pad to be released.

## Using the Mellosoftron as a virtual MIDI device

There are several software synthesizers on the market which act as virtual MIDI devices. After installation, they look just like an ordinary MIDI output port connected to an external sound module (to other programs running on the same system, that is).

While this kind of setup makes sense for a "black box" soft synth with a small, proprietary sound library which can (and must!) be kept in memory at all times so as to emulate a low-end General MIDI module, it isn't really adequate for a full-blown, programmable sampler like the Mellosoftron, which you probably don't want to keep up and running at all times.

Having said that, it is nevertheless quite easy to let the Mellosoftron act as a virtual MIDI device serving a sequencer or a MIDI player running on the same system.

The trick is to install one of several freely available virtual MIDI port drivers. Select the virtual port as the Mellosoftron's MIDI input port and as your sequencer's or MIDI player's output port, and you're all set.

Unlike the Mellosoftron, such a driver has a negligible memory footprint and will not use up any valuable CPU time while it's idle. You can therefore leave it permanently installed and launch the Mellosoftron only as needed.

As for the choice of virtual MIDI port driver, there are several floating around on the Net, but an excellent one is [Hubi's MIDI LoopBack Device 2.4](#) (or later - earlier versions did not work with 32 bit programs like the Mellosoftron) by Hubert Winkler. A painless 43 kB download available at all the large software archives (SimTel, WinSite etc.), it's not only free - it even comes with full source code! Look for the file [mdlpbk24.zip](#) in the [Windows 3.x](#) sound areas (it works under Windows 3.1 too, so it's archived there) or at the Mellosoftron's own official download site,

[www.polyhedric.com/software/ml/download.html](http://www.polyhedric.com/software/ml/download.html)

Unzip into an empty directory, follow the installation instructions in [midlpbk.txt](#) and you will have it up and running in no time at all.

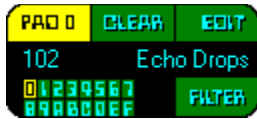
A last tip. For the reasons stated above, this is **not** recommended, but if you really love the Mellosoftron so much that you want to keep it up and running at all times (maybe serving as an emulated General MIDI / GS module) here's how you can do it:

- Right-click the taskbar's **Start** button. Click **Open**.
- In the **Start Menu** folder, right-click the **Programs** folder. Click **Open**.
- In the **Programs** folder, right-click the **StartUp** folder and (you guessed!) click **Open** again. This is your destination - anything in the **StartUp** folder is executed automatically when Windows is started, and that's what you want to be done with the Mellosoftron.
- Open a Windows Explorer view on the directory where you installed the Mellosoftron (if you can't remember where that is, run **Find** from the **Start** menu and let it look for the file **ml.exe**). Select the file **ml.exe** (the main Mellosoftron executable).
- Drag **ml.exe** from the Windows Explorer window to the **StartUp** folder and drop it there.
- Close all open folders and Windows Explorer.

- Launch the Mellosoftron. Minimize it.
- On the taskbar, right-click the Mellosoftron's button (**don't** restore the Mellosoftron by left-clicking its button - if you do so by mistake, minimize it again and repeat this step). Click **Close**. The next time it's run, the Mellosoftron will remember that you want it to be minimized.
- Restart Windows. When the desktop is done repainting itself, you will see the Mellosoftron appear briefly and then minimize itself. This will happen every time you start up Windows, until you drill down to the **StartUp** folder again and delete the Mellosoftron's shortcut. (Actually, the Mellosoftron will always start up in its last window state, whatever that was - minimized, normal or maximized. If you restore it before shutting it down, the next time it will come up as a normal window. Just minimize and close it again to force it back to minimized operation.)

## Editing an existing instrument patch (PRG)

The instrument patch (PRG) editor is invoked by clicking the **Edit** button in the upper right corner of any active instrument pad:



If all pads are free, you can either load or create a new PRG to edit.

Since PRGs are just structured text files, they can actually be modified using any text editor. Their format is not free though, so you should use a dedicated editor with them unless you have very good reasons not to do so.

The Mellsoftron's PRG editor is a modeless window. You can therefore switch freely between it and the main window.

You don't have to content yourself with just one editor window either. All 16 pads can have their own PRG editor window open at the same time, each one on a different PRG. If you like cluttered screens, here's your chance to party - but try not to go crazy just because you can!

Now for the nitty-gritty...

- [The PRG editor: an anatomy lesson](#)
- [PRG editing step by step](#)

## The PRG editor: an anatomy lesson

[Click [here](#) for an introduction to PRG editing.]

The Mellosoftron's PRG editor is essentially three different editors rolled into one:

- The window's upper panel contains a spreadsheet-like data grid which can be used to edit all parts of a PRG.
- The lower left panel contains a graphic envelope editor.
- The lower right panel contains a combined WAV viewer and sample position editor.

All three panels can be resized by dragging the splitters which separate them.

## Generic principles

Each PRG editor window is owned by an instrument pad.

When a new PRG editor is opened, it reads the PRG file assigned to the owning pad and puts it into a private edit buffer. **None of the changes made to the edit buffer are reflected in the PRG file until you click Save (or press Alt-S).** Doing so causes the edit buffer to be checked for correctness and, if no errors are found, to be saved back to disk using a standard Save File dialog (allowing you to change the filename if you don't want to overwrite the source file).

If there are errors in the edit buffer when you try to save it, the Mellosoftron won't open the Save File dialog. Instead, it will highlight the grid location where it got confused and tell you what it doesn't understand. When this happens, keep in mind that the cell pointed out by the Mellosoftron may or may not be the location of the actual error - especially envelope and loop positions are interrelated, and can be inconsistent for several reasons. If a loop start position trails the loop end position, which one is wrong?

Clicking **Reload** (or pressing Alt-R) causes the PRG file to be re-read into the edit buffer from disk. If you do this without saving first, all changes you made to the edit buffer will be lost.

Closing the editor window without saving does not cause the edit buffer to be lost, on the other hand. As long as the pad isn't cleared (by loading a new PRG, for instance), you can resume editing at any time by clicking the pad's Edit button again.

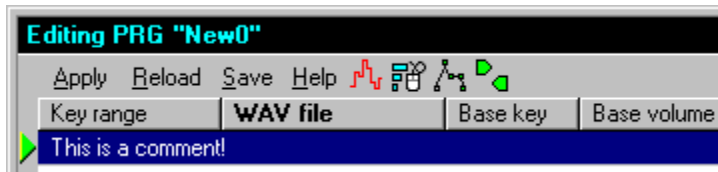
**None of the changes made to the edit buffer have any effect on the owning pad until you click Apply (or press Alt-A).** Clicking Apply causes the edit buffer to be checked for correctness and, if no errors are found, to be copied to the pad, allowing you to play the modified PRG as usual. If there are errors, the pad's settings are left untouched. Instead, as when trying to save to disk, the Mellosoftron will highlight the grid location where it got confused and tell you what it doesn't understand.

When the edit buffer is successfully copied to the owning pad, the Apply button turns into a **Revert** button. Clicking Revert (or pressing Alt-V) switches the pad back to its previous settings and turns Revert back into an Apply button. This makes it easy to toggle back and forth between the original and the modified PRG until you are satisfied that the latter is actually an improvement over the former.

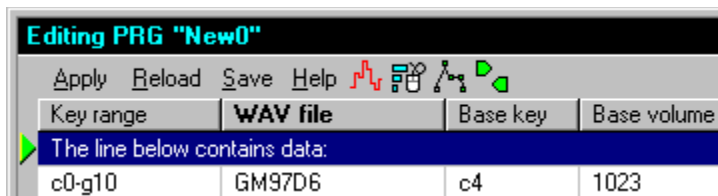
## The data grid

The data grid closely reflects the structure of PRG files. A PRG file consists of rows, each one containing either a free text comment or data. The meaning of that data is addressed in the section on [PRG editing](#). Here, we are primarily concerned with the tools used to manipulate it.

Comment rows are displayed as a single cell spanning all grid columns:

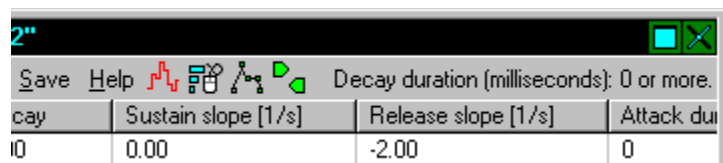


Data rows are displayed and edited column-wise:



To edit a cell, you must first select it, then click it again or press Enter to open its in-place editor. The kind of in-place editor you'll get depends on the cell's content. Comment cells are edited in plain text edit boxes. Most numeric data values are edited in text edit boxes augmented by spin buttons, while some data fields have more sophisticated in-place editors. What they all have in common is that you can change your mind and discard your edits by clicking Esc, or commit to the new values (and close the in-place editor) by double-clicking the in-place editor or pressing Enter.

You can move through the grid using the mouse, the keyboard's arrow keys and the scroll bars. In addition, the four speedbuttons on the menu bar scroll the grid horizontally to the first column in the WAV, controller, envelope and sample position sections, respectively. The rest of the menu bar is used to display a description of the first (upper left) data cell in the current selection (if any):



Note that multiple cell selections are allowed. This is primarily useful for moving, copying and deleting cells.

To move one or more rows, select at least one cell in each row you want to move and drag the row marker displayed on the grid's left side to the new position:

f3-g10	GM38D#3	d#3	96
e3	SNARE2	e3	511
d#3	CLAP	d#3	1023
c#3	SNARE1	d3	511
c#3	HIGHQ	c#3	1023
c#3	KICK1	c#3	384

Note that clicking the panel around the grid's row marker causes the corresponding row to be selected.

To delete one or more rows, select them, right-click the grid to bring up its pop-up menu,

SNARE2	e3	511
CLAP		1023
SNARE1		511
HIGHQ		1023
KICK1		384
KICK2		384
CLAP		511

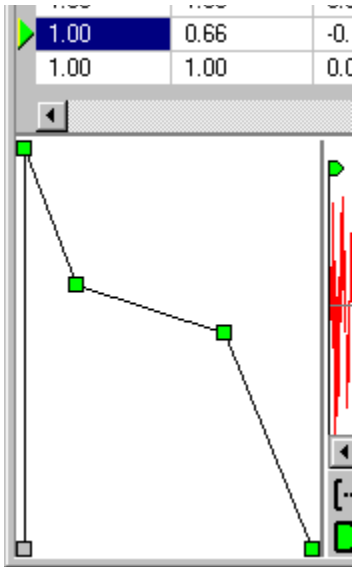
and click **Delete rows**. To duplicate the selected rows, use either **Insert duplicate** in the **Insert...** submenu or **Add duplicate** in the **Add...** submenu (the only difference is where the duplicates end up, above or below the source rows). These submenus also let you insert and add new comment and data rows.

The **Copy to...** submenu is more tricky: it only affects data entries (not comments) in the selected columns, and is used to copy selected cells to other cells in the same columns, subject to conditions like the WAV file referenced by source and target row being identical. You will understand its use after reading the section on [PRG editing](#).


## The envelope graph

The Mellosotron lets you impose classic ADSR (Attack-Decay-Sustain-Release) envelopes on samples. You can edit all parameters affecting envelope shape in the grid columns starting with **Attack**, but you can also work with a graphic representation of the same parameters:





The envelope graph displays the envelope of the top row in the grid's current selection (if any).

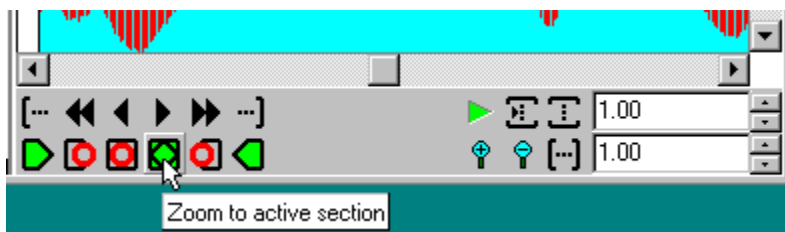
The easiest way to understand how modifying the envelope graph affects envelope parameters is to click  in the PRG editor's menu bar and then watch the values in the grid change as you drag the envelope markers around. You can also edit the values in the grid and watch the effect on the envelope display (but you'll have to close the grid's in-place editor every time you want the graph to be updated).

Note that the leftmost marker in the envelope display can not be moved, as all envelopes start at zero amplitude, and that the rightmost marker can only be moved vertically, as the time scale is set automatically for you: the sustain and release sections describe slopes, not absolute time and amplitude values (those must be computed dynamically when each note is played, since they depend on the duration and velocity of the note event).

## The WAV graph

The WAV graph displays the WAV file referenced by the top row in the grid's current selection (if any).


With one exception, all controls below the waveshape display are used to scroll around the sample data and to zoom the display. If in doubt, place the mouse cursor above the control you're interested in and read the hint that pops up after a second or two:



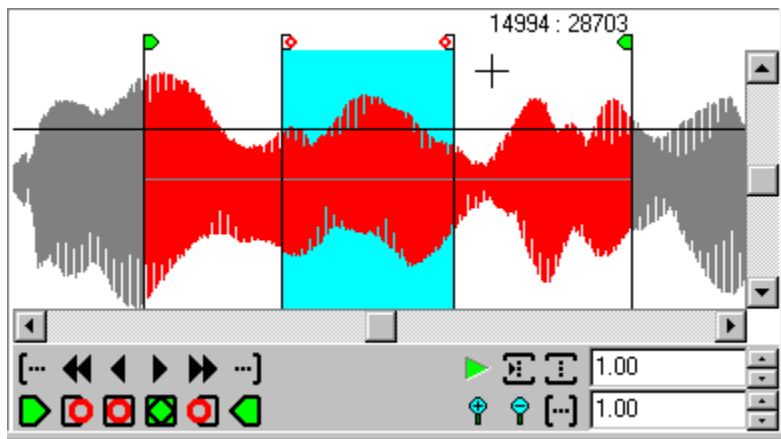
Note in particular that the button row in the lower left corner can be used to quickly scroll to the **Start sample**, **Loop start sample**, **Loop end sample** and **End sample** positions set

in the data grid, as well as to zoom to the active section (Start sample to End sample) or to the loop section (Loop start sample to Loop end sample).

The two edit boxes can be used to set amplitude (top box) and time zoom factors (bottom box). A time zoom factor 1.0 means that the entire WAV is shown, an amplitude zoom factor 1.0 that the full dynamic range supported by the WAV's format (0 to 255 for 8 bit WAVs, -32768 to 32767 for 16 bit WAVs) is shown.

The exception mentioned above is the  (play) button. Use it to play the active WAV section. If loop start and end positions are set, playback is looped accordingly. The grid's envelope settings are not used.

Here is an example of what the waveshape display might look like:



A few highlights:

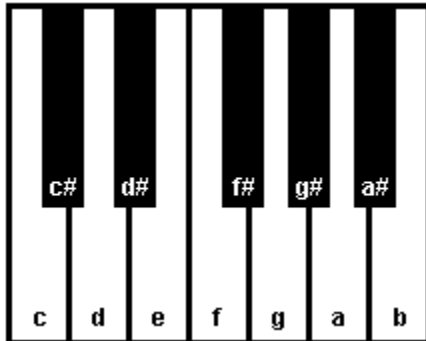
- The current cursor position is displayed above the waveshape display as "sample number : vertical position". Note that it slides horizontally to follow the cursor.
- WAV sections preceding the Start sample position and trailing the End sample position are grayed out.
- The loop section is highlighted (aqua-colored background).
- Sample positions are marked by small icons above the waveshape display. Compare the symbols with those on the buttons in the graph panel's lower left corner. These icons can be dragged to change the sample positions. All changes are immediately reflected in the data grid.
- If a sample position is selected in the data grid, clicking the graph causes that position to be set accordingly. If it's a Loop start sample or Loop end sample position, a vertical line will also be drawn in the graph, marking the vertical position you clicked. This makes it much easier to find samples with similar values (useful when looking for loop points).
- Changes in the data grid are reflected in the waveshape display as soon as they are confirmed by double-clicking the grid's in-place editor or by pressing Enter.

See the section on [PRG editing](#) for details on how all this is used!

## PRG editing step by step

The basic layout of PRG files is quite simple. Each line applies to a key range, which may be as wide as the whole keyboard (c0 to g10) or as narrow as a single key. The line assigns a WAV file to the key range and describes how it should be played - which envelope to use, whether (and if so how) to loop it, how to handle controller events and so on.

Keys are described in standard notation: a letter (followed by a "#" for black keys) and an octave number.



Key ranges can overlap. When this happens, pressing a key will cause all WAVs assigned to it to be played together, each one with the envelope and performance parameters described by its own PRG line.

Synth-heads call this layering. It can be a very powerful technique. Layering slightly detuned WAVs will give you a richer, "fatter" sound. Layering instruments which react differently to MIDI controllers (e.g. attack and release velocity) can yield spectacular dynamic response.

The MIDNight Express has a six-layer limit (no more than six separate PRG lines on each key). The Mellosoftron III and WAVmaker III have no such hard-coded limitation, but the Mellosoftron can still run out of voices - each layer is played by a separate one.

So, how do you create a PRG? That's easy: you copy an existing one - preferably one using the same keyboard mapping - and edit it.

The choice of keyboard mapping should be made as soon as possible, and depends on several considerations. Which samples are available? How large is the patch (with samples) allowed to get? Which software are you going to use with your new PRG?

Most PRGs in the GSound 22 library and in WAVmaker's GSound 44 library use sample files recorded at f3, f4, f5, f#3, f#4, f#5, and often also at f6 and f#6. All other notes are obtained from these by resampling. The reason for having two adjacent base notes for each sampled octave is essentially mathematical: everything else being equal, resampling by harmonic ratios (such as f to g or f# to g#) yields better sound quality (less distortion) than resampling by inharmonic ratios (such as f to g# or f# to g). For even higher quality, one can use the mapping employed for GSound's acoustic pianos: in this case, the full octave is obtained from d, d#, g# and a, by moving one note up or down from the appropriate base note.

Sound quality and speed are the opposite sides of the same coin. The Mellosoftron and the MIDNight Express use linear interpolation for fast resampling. For them, moving up from the base note (e.g. from c4 to c5) is quite cheap (in terms of CPU time) but leads to quicker deterioration in sound quality. Moving down (as in c4 to c3) is expensive, since more and

more data points have to be interpolated between existing ones - moving down by one octave requires an interpolated sample to be computed for each recorded one - but quality doesn't deteriorate as fast.

The simplest option is to work "MOD-style", i.e. sample just one note, like c4, and derive all other notes from it. This only takes one PRG line, in which you declare the **Key range** to be c0-g10 (more generally, remember that contiguous ranges sharing the same base key and other parameters need not be described key by key). The problem with this approach is that most instruments (especially acoustic ones) do not only vary in pitch but also in timbre (i.e. in waveshape) when played at different notes (just think of an acoustic piano being played at opposite ends of the keyboard). In order to reproduce this kind of behaviour, you'll need to spread out your base keys over a suitable range.

Having said that, in the example below we are nevertheless going to use the trivial mapping with just one base note, since it's sufficient for showing (almost) all the steps involved. Creating additional lines and/or layers is just more of the same.

Our starting point will be the default PRG created by the Mellosotron's own **New** buttons.

- **Step 1:** Follow the standard procedure for [creating a new PRG](#).

Use the sample file GM97D6.WAV (included in the Evaluation Package).

We could also have [loaded an existing PRG](#) to use as our template. As noted above, this can be a better option when dealing with more complicated keyboard mappings (if there is an adequate PRG to start from, that is).

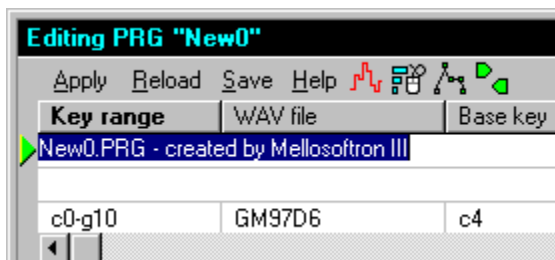
- **Step 2:** Click the pad's **Edit** button.

This causes your new PRG to be loaded into a [PRG editor](#) window.

- **Step 3:** Edit the comment line.

When the PRG editor starts up, the first PRG line is automatically selected. It happens to be a comment, since the Mellosotron always puts a comment first in new PRG files. It's usually a good idea to replace this default comment ("created by Mellosotron III") with something more descriptive. This way, there will be no risk of confusion when you return to the file later ("What is this patch for? Do I need to keep it?").

Click the comment line or press Enter to open its in-place editor:



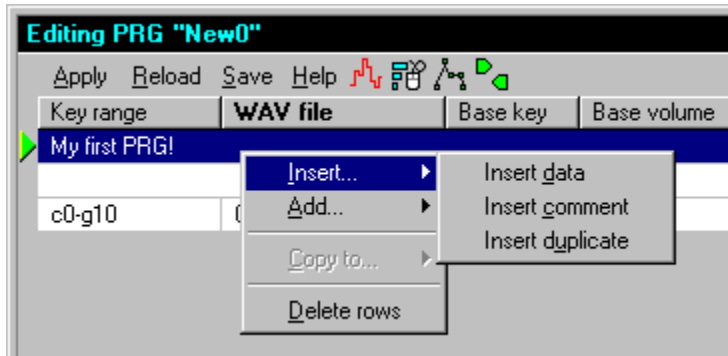
Change the text to something like

My first PRG!

and press Enter again to close the in-place editor.

Text entry works the same way throughout the data grid: click Esc if you change your mind after editing a value in the in-place editor, double-click or press Enter to commit to the changes and close the in-place editor.

If you want to enter more comment lines, you can right-click the grid to bring up its pop-up menu:



Choose **Insert comment** to create a new, empty comment line above the current one; choose **Insert duplicate** to insert a copy of the current selection. **Add comment** and **Add duplicate** work quite analogously, but create the new line(s) below the current selection.

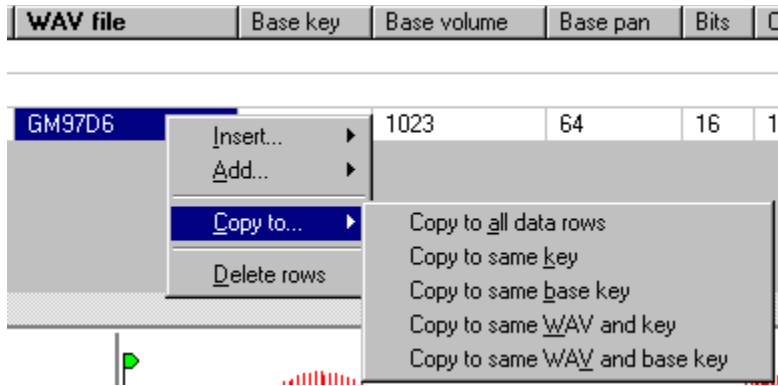
- **Step 4:** Change sample file(s) (if needed).

This step is not actually needed in our example. After all, we are using a single WAV as our sound source, and were therefore able to pick it already when we created the new PRG. But if we were using an old PRG based on different WAVs as a template (in order to inherit its keyboard mapping and/or other properties) this would be an important part of the process.

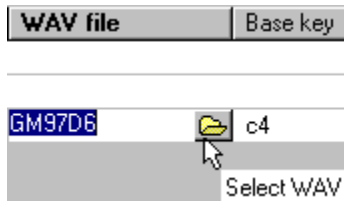
Depending on your goals and on the file at hand, you may prefer to use a text editor when replacing WAV file references. Remember, PRG files are just structured text files, so they can be edited that way too.

Suppose that you have a PRG referring to a large number of different WAV files. The 6.PR.G (Harpisichord) contained in WAVmaker's GMega Sound Library, for instance, refers to WAV files called GM6F3, GM6F#3, and so on. If you want to change those references to (e.g.) the string sample files GM48F3, GM48F#3, ..., you can use a text editor to perform a global "search and replace" operation, from GM6 to GM48, with just a couple of keystrokes. This is also worth keeping in mind when doing other large-scale PRG editing.

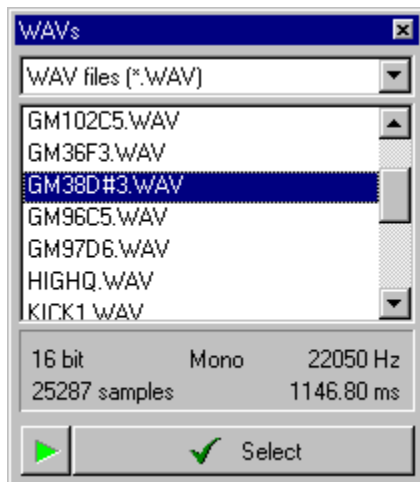
On the other hand, the dedicated PRG editor can do some things which a text editor can't do, since it knows the meaning of the data. The **Copy to...** menu can be used to perform selective copy operations to data lines sharing the same key (useful with layered sounds), the same base key, the same WAV and key or the same WAV and base key. It can also perform a global copy operation:



The **WAVs** window can also come in handy. To bring it up, select the **WAV file** cell, click it again or press Enter to open its in-place editor, then click the in-place editor's folder button:



You should now be looking at the WAVs window:



Use it to browse and play the sample files in the current WAV directory. Mark a filename and click the **Select** button to cause the filename to be copied into the PRG editor.

You can of course also type the WAV filename directly into the in-place editor.

If you want to add more WAVs, you can either use the grid's pop-up menu to insert/add data lines or (even faster) drag the WAV files from the Windows Explorer and drop them on the PRG editor.


The End sample value is updated automatically when a WAV reference is added or changed; the Start sample, Loop start sample and Loop end sample values are reset.

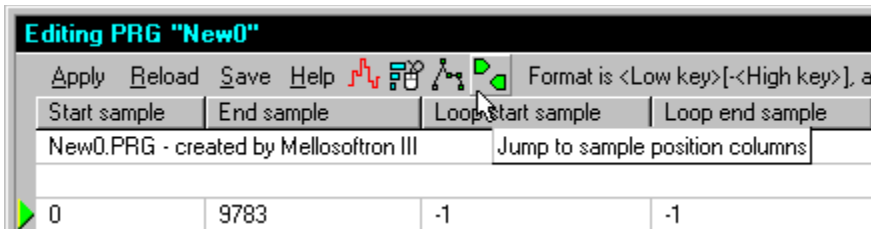
Remember that the WAV file extension is always implied - whether you are editing your PRG with the Mellosotron's dedicated PRG editor or with a text editor, don't type it in!

- **Step 5:** Find new loop points.

As anybody who has ever been involved with sampling will tell you, this can be a most frustrating task! You need to find two samples with similar (ideally, identical) values, separated in time by a multiple of the sound's wavelength. Since any reasonably interesting sample file will feature large envelope and timbre variations over time, you'll be tempted to use "tight" looping, i.e. to constrain the loop to a small region within which it's relatively easy to find corresponding samples - but if you do this, you'll suppress exactly those variations which make the sound interesting. You may get away with tight looping when the loop is used to extend a low-volume release section, but in most cases, you'll use looping for the sustain section, which is central to how the sound is perceived.

Your best tool in this game is the PRG editor's WAV graph and its scaling/zooming controls. Here's what you do in a typical looping session:

- **Step 5.1:** In the data grid, select the line on which you want to work.
- **Step 5.2:** Click  to scroll the grid horizontally to the sample position section:



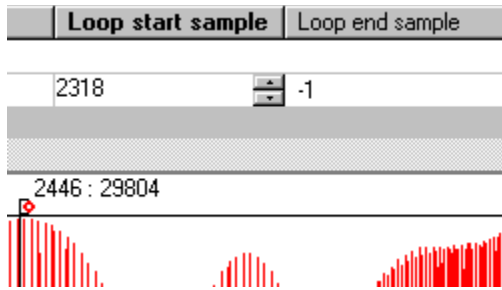
- **Step 5.3:** In the WAV graph panel, click  to change the time zoom to 1.0 (i.e. to display the whole WAV). This is useful for getting a first impression of the WAV and for identifying promising loop sections.

(Note that the WAV graph zooms automatically to the active WAV section, i.e. to the section located between the Start sample and the End sample positions, when you select a new data line in the grid. Since the Start sample and End sample positions are set to include the whole WAV whenever a WAV reference is modified, step 5.3 is not really necessary in our example; the entire WAV is already displayed.)



- **Step 5.4:** In the data grid, select the **Loop start sample** cell. Optionally open its in-place editor by clicking the cell again or by pressing Enter:




- **Step 5.5:** In the WAV graph, click a tentative loop start position. Several things will happen:



- A crosshair is added to the graph, marking both the horizontal (time) and the vertical (amplitude) position you clicked.
- A small icon denoting the loop start position is added above the waveshape. You can move the loop start position by dragging this icon. Click the graph again if you prefer to change the marked amplitude too.
- The loop start position is updated in the data grid. You can use the in-place editor to adjust it. As usual, click Esc if you change your mind after editing a value in the grid, double-click the in-place editor or press Enter to commit to the changes and update the WAV graph accordingly.

- **Step 5.6:** Use the  ("Go to loop start") button, the  ("Zoom in") button and the time zoom edit box to successively zoom in on a good loop start position. Some people will tell you that zero crossings are good loop points, but peaks and valleys are often better, since they are (near) zero-slope points, i.e. the waveshape is only changing slowly there. As a consequence, you're more likely to find at least one good fit at the opposite end of the loop section if you use peaks and valleys.

- **Step 5.7:** When you're satisfied with the selected loop start position, click  to zoom out again.
- **Step 5.8:** In the data grid, select the **Loop end sample** cell. Optionally open its in-place editor by clicking the cell again or by pressing Enter:

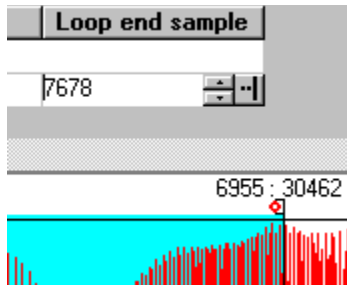




- **Step 5.9:** Use the crosshair to find a good fit (same amplitude) for the end of the loop. Click it.


(If you can't find a good fit, go back to Step 5.4 and pick another loop start sample. Yes, life is hard.)

Again, several things will happen:





- The selected loop section is highlighted in the WAV graph.
  - A small icon denoting the loop end position is added above the waveshape. You can move the loop end position by dragging this icon or (as long as the Loop end sample is selected in the data grid) by clicking the graph again.
  - The loop end position is updated in the data grid. You can use the in-place editor to adjust it. As usual, click Esc if you change your mind after editing a value in the grid, double-click the in-place editor or press Enter to commit to the changes and update the WAV graph accordingly.
- **Step 5.10:** Use the  ("Go to loop end") button, the  ("Zoom in") button and the time zoom edit box to successively zoom in on a good loop end position (see step 5.6).

- **Step 5.11:** Listen to the loop using the  button. Be sure to listen long enough for the loop to be traversed a few times over. Adjust the loop start and loop end positions in small steps until there are no audible clicks. If you can't get rid of the clicks, go back to Step 5.7 or (in desperate cases) to Step 5.3.

- **Step 5.12:** [PRGs with multiple data lines only] In the data grid, use the pop-up menu's **Copy to...** commands to update all loop start and loop end entries for the same base key (same WAV and base key if you are layering different WAVs).

- **Step 5.13:** Click **Apply** (or press Alt-A) to make the modified PRG take effect and check that the loop also sounds right for the resampled notes.

If it doesn't, you may want to supply different loop points for the offending notes only rather than start all over again. Use the pop-up menu's Add and Insert commands to create separate lines for these notes, adjust their loop points in small steps, re-Apply and re-check until the clicks are gone.

- **Step 5.14:** Click Save (or press Alt-S) to save the new PRG.

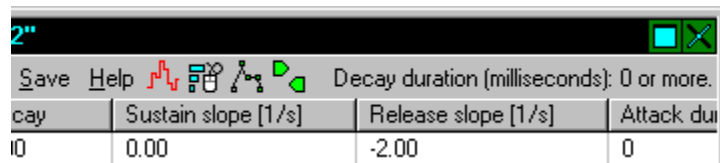
**IMPORTANT: the PRG file is not saved automatically!**

If you think this is a lot of work, imagine what it was like to create a couple of GS libraries! Anyway, try looping GM97D6. When you get too sick of it all, you can look at 97.PRG for a suggestion.

Cheer up, though. After all, not all sounds need to be looped!

**Step 6:** Edit the other performance parameters.

There are quite a few of them, and no real need to go through them all in detail here. You can find out what they do by selecting them and reading the description line in the PRG editor's upper right corner:



The screenshot shows a window titled "2" with a menu bar containing "Save", "Help", and several icons. Below the menu bar, there is a text field with the label "Decay duration (milliseconds): 0 or more." and a value of "0". Below this is a table with four columns: "cay", "Sustain slope [1/s]", "Release slope [1/s]", and "Attack duration". The table contains one row of data.

cay	Sustain slope [1/s]	Release slope [1/s]	Attack duration
10	0.00	-2.00	0

Experiment a bit and you will understand them in no time at all. A few highlights:

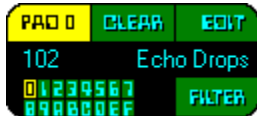
- Raise the **Base key** (the note which you want the Mellosotron to believe is played by the WAV) in order to lower the pitch at which it's played back.
- Raise the **Base volume** (the volume at which you want the Mellosotron to believe that the sound was recorded) in order to reduce playback volume. (Read the last statement again!)
- Raise the **Base pan** (the stereo pan position at which you want the Mellosotron to believe that the sound was recorded) in order to have it play back further to the left.
- Set the **Start sample** to a value larger than zero if you want to skip a leading section of the WAV file.
- The **End sample** can be used to limit the max duration of un-looped sounds. If the sound is looped, **End sample** must be equal to or larger than **Loop end sample**.
- Use the **Attack**, **Decay**, **Sustain slope**, **Release slope**, **Attack duration** and **Decay duration** fields to modify the envelope shape. These should all be fairly intuitive. Note that you can edit the envelope graphically, too.
- Modify the **Attack sens.**, **Pressure sens.**, **Expression sens.**, **Release sens.** and **Mod. Sens.** settings to get more or less dynamic response. Note that negative values are allowed for all but the modulation sensitivity, so you can have (for instance) instruments that are played more silently the harder you hit a key! Together with layering, this feature allows you to create instruments with controller-dependent timbre (mix different layers in proportions determined by attack velocity etc.).

## Creating a new instrument patch (PRG)

New instrument patches (PRGs) are created by clicking the **New** button in the upper right corner of any instrument pad with no patch assigned to it:

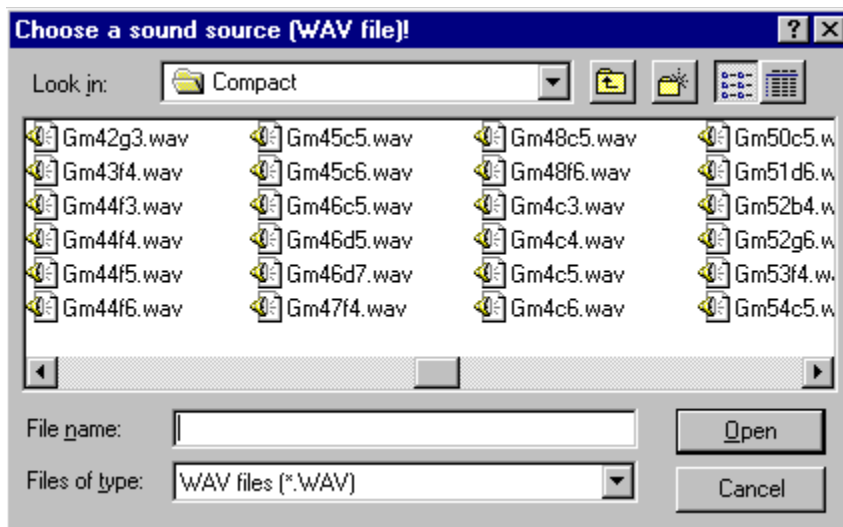


If all pads are taken, just pick one and click its **Clear** button:



This causes the PRG on the pad to be unloaded.

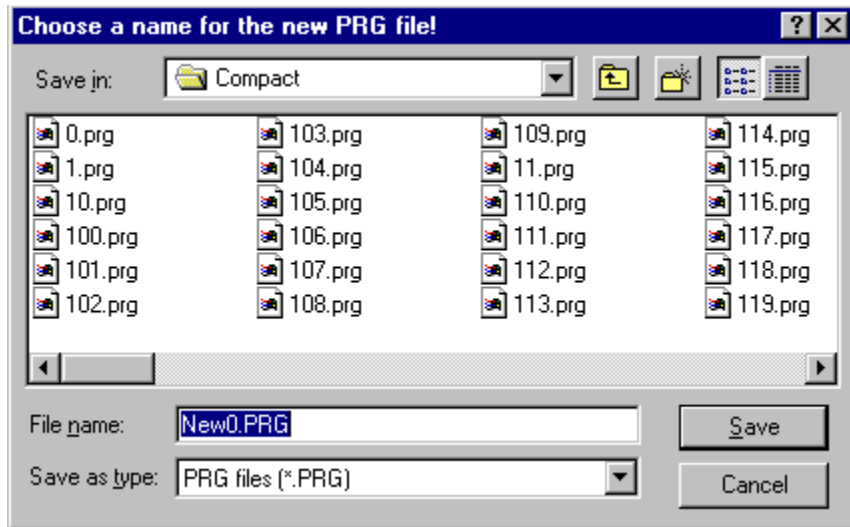
Click the **New** button! You will be presented with a standard Open File dialog giving you a view of the current WAV directory:



Select a WAV file and click **Open**. The Mellosotron will check the WAV header and, provided that no problem is found with it, display the current PRG directory in a Save File dialog prompting you for a name to use for the new PRG. The default name is "NewX", where X is the pad number.

You can also create a new instrument by dragging a WAV file e.g. from Windows Explorer and dropping it on the Mellosotron's main window. The new instrument will be created on the first free instrument pad.

Here, we are creating a new PRG on instrument pad 0:



Click **Save**. The new PRG is created and - provided that the WAV loads OK - assigned to the pad, which is then activated.

You can now play and edit your new PRG as usual.

## About MAPs

The Mellosoftron's 16 pads allow you to work with up to 16 different instrument patches (PRGs) at the same time. Since you probably don't want to reload a bunch of PRGs manually every time you start the Mellsoftron, it takes care of bookkeeping what's on each pad and lets you load and save MAP files containing this information at the click of a button.

MAPs also contain MIDI MAPPING and filter status (filter enabled/disabled) information.

The Mellosoftron looks for and stores MAPs in the default MAP directory. Click the **MAPs** button in the main window to change MAP directory.

In order to load a MAP, click the **Load** button in the main window's upper left corner:



You will be presented with a standard Open File dialog. Select a MAP file and click **Open**. The Mellosoftron will read the new instrument assignments and update its pads accordingly. Pads without an instrument assignment in the MAP being read will not be affected.

As of version 2.3, you can also load MAPs by dragging them from e.g. Windows Explorer and dropping them on the Mellosoftron's main window.

Loading a MAP causes it to become the new **default MAP**, i.e. the MAP which is loaded automatically at the beginning of the next session.

The main window's **Merge** button allows you to read a MAP file without first clearing the current assignments, and without changing default MAP.

Use the **Save** button to store MAPPING information. The Mellosoftron will ask you if you want to save the current MAP before loading and merging, as well as at shutdown. Saving a MAP causes it to become the new default MAP, just like loading.

## MIDI MAPping

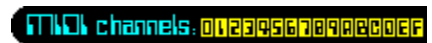
Since the Mellosotron has 16 instrument pads, and 16 also happens to be the number of MIDI channels, it's natural to expect MIDI channel 0 to control pad 0, MIDI channel 1 to control pad 1, and so on.

This is indeed the default situation - but far from the only possibility. The Mellosotron allows you to map any combination of MIDI channels to any instrument pad, to select active MIDI channels and to treat any MIDI channel as a drum channel (see below). These settings are stored in the same MAP files which also control filter status and assign instrument patches (PRGs) to pads, allowing you to store and quickly retrieve complete performance setups.

The only pieces of MIDI setup information which do not go into MAP files are the MIDI port selection, which is stored separately so as to avoid unnecessary opening and closing of MIDI ports (after all, you are unlikely to want to use different ports with different performance setups), and Omni On/Off status (see below).

## MIDI message routing

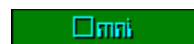
All MIDI messages coming in from the MIDI input port are matched against the Mellosotron's global MIDI channel settings. Only messages on active channels, i.e. channels which have been highlighted in the global MIDI channel selector immediately below the instrument pads,



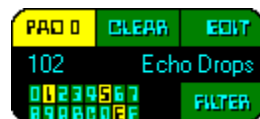
are let through. In the picture above, all MIDI channels are active.

If a MIDI message makes it through this first step, there are two possibilities:

When **Omni** mode is off,



the message is forwarded to all instrument pads. Each pad then matches the message against its own MIDI settings. For instance, this pad will respond to messages on MIDI channel 1, 5 and 14 (hexadecimal E):



The pad will ignore all messages on other channels.

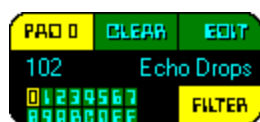
When Omni mode is on,



the message is only forwarded to the selected instrument pad, and the pad's own MIDI settings are ignored to let all messages through, no matter which channel they are on.

Filter control messages (resonance and cut-off, 47H and 4AH respectively) are also matched against the pad's filter status. In the pad picture above, they would be ignored. In the picture

below, the pad is set to respond to filter control messages:



## Drum channels

Program Change messages are routed like all other MIDI messages, but their interpretation is also affected by the global drum channel selector:



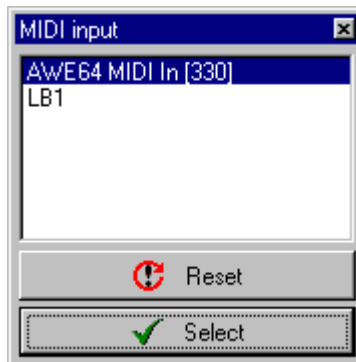
In the picture above, MIDI channel 9 is the only drum channel, as in the GM/GS standard (don't be confused by the Mellosoftron's use of 0-based channel numbering; chances are that you are used to thinking of this channel as number 10).

Any pad set to listen to a drum channel will interpret all Program Change messages it receives as referring to drum sets, including messages not originating from a drum channel. You could say that drum channels are "dirty": it's enough for one of the pad's input channels to be a drum channel for the whole pad to become a drum pad.

When the drum status of a pad is modified, either by adding a drum channel to a non-drum pad or by removing all drum channels from a drum pad, the pad is cleared. If the name of the PRG being discarded conforms to the WAVmaker/MIDnight Express naming convention based on MIDI program numbers, an attempt is then made to load the corresponding PRG from the alternate set. For instance, 0.PRG would be replaced with D0.PRG (if present in the current PRG directory) and vice versa.

## MIDI input

Click the **MIDI** button in the Mellosotron's main window to bring up the **MIDI input** dialog:



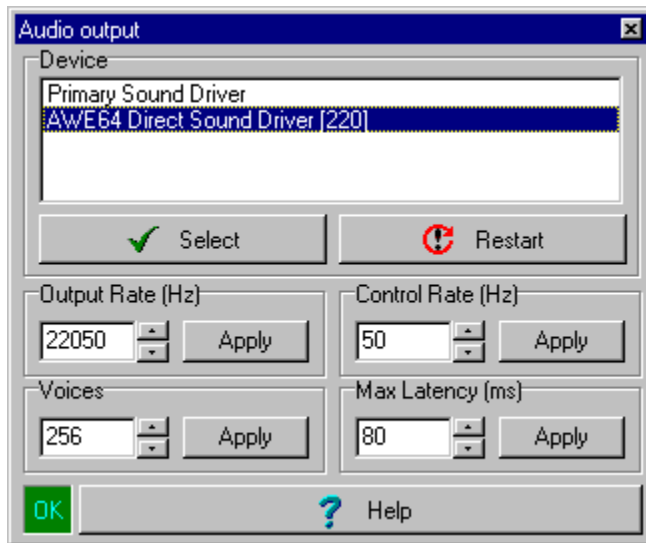
To change MIDI port, select it in the list and click **Select**. The Mellosotron will switch to the new port immediately. If there is a problem, it will warn you and try to revert to the previous port, otherwise it will just keep going without further ado.

The **Reset** button does just what it says: it resets the MIDI port, and might come in handy if you experience MIDI-related problems.



## Controlling audio output

Click the **Audio** button in the Mellosoftron's main window to bring up the **Audio output** dialog:



The controls in the **Device** box work in a way similar to their counterparts in the [MIDI input](#) dialog:

- To change output device, select it in the list and click **Select**. The Mellosoftron will switch to the new device immediately. If there is a problem, it will warn you and try to revert to the previous device, otherwise it will just keep going without further ado.
- The **Restart** button stops and immediately restarts audio output. This is essentially a (very) mild "reset" which you can try using if the sound doesn't come out right after changing some setting.

In addition to selecting and restarting an output device, you can also...

- Set the **Output Rate**. This is the sampling rate of the audio produced by the Mellosoftron.

The default rate of 22050 Hz is the same used by the MIDInight Express, the Roland Virtual Sound Canvas and other soft synths, and should be adequate for most purposes. You can get a better high-frequency response by raising this value, provided that your instrument samples are sampled at the higher rate too - if not, you are just wasting CPU time. CD quality is 44100 Hz, and the Mellosoftron will go as high as DirectSound will let it, i.e. 100000 Hz.

You must click the **Apply** button next to the input field for the new output rate setting to take effect.

- Set the **Control Rate**. This is the rate at which envelope factors are updated.

If you don't know what that means, don't worry - you will, once you have [edited your first PRG](#). PRGs (instrument patches) allow you to specify classic ADSR (Attack - Decay - Sustain - Release) envelopes, which are linearly interpolated on the fly to yield actual amplitude factors (since the note duration and other controller values are not known

before a note is actually played, envelope factors must be computed at runtime - there just is no alternative).

A higher **Control Rate** means that the envelope factors are updated more often (which in turn means a higher CPU load). In principle, the **Control Rate** can be as high as the **Output Rate**, meaning that envelope factors are recomputed for each sample in the output stream. In most cases, this would be a total waste of CPU time. The default setting of 50 Hz is usually sufficient.

You should only change the **Control Rate** if you have good reasons to do so (typically steep envelopes applied to short notes, requiring a higher setting for smooth interpolation).

You must click the **Apply** button next to the input field for the new control rate setting to take effect.

- Set the number of **Voices**. This is the number of "oscillators" (monophonic voices) available to the Mellosotron.

Do not confuse the number of voices with actual polyphony! The Mellosotron might have 256 voices but only enough CPU power to play (say) 30 of them at the same time. So why all the "extra" voices? Because voices are not grabbed and released instantly when notes are turned on and off, but rather with a delay ranging up to the **Max Latency** setting. If this setting is 80 ms, a voice actually producing sound for only 40 ms could remain unavailable to new events for as long as 160 ms. This is a worst-case example (CPU just barely keeping up with the load and a very short note) but it illustrates how a few extra voices can come in handy.

This value should probably not be changed at all. Unused voices have a negligible effect on the CPU load and a very modest memory footprint, so there isn't much to be gained from getting rid of them. If you nevertheless want to change the number of voices, remember that you must click the **Apply** button next to the input field for the new setting to take effect.

- Set the **Max Latency**. This setting determines how long the Mellosotron is allowed to take (in milliseconds) before producing an audible result from an event (e.g. a key press). [This is a worst-case setting](#). The actual latency is normally 1/4 of the value given here, so 80 ms corresponds to a normal latency of 20 ms. This is as low as DirectSound will go, and below the human perception limit. The Mellosotron will take longer only under exceptional circumstance, e.g. CPU load peaks. If such a peak ever introduces a delay longer than the **Max Latency** setting, the audio stream will break up.

You should touch this value only if you are already hearing breakups and are convinced that the cause is the CPU not keeping up (a good way to find out is to run the Windows System Monitor, Programs | Accessories | System Tools | Sysmon, alongside the Mellosotron). In this case, raising the Max Latency just might get rid of the breakups, at the cost of responsiveness (higher latency results in lower CPU load). Don't do this just in order to play complex MIDIs - it's much better to use the [MIDInight Express](#) for that!

You must click the **Apply** button next to the input field for the new max latency setting to take effect.

Finally, keep an eye on the status field in the dialog's lower left corner when changing audio settings. It reports the result of the latest operation. In case of errors, the first thing to try is clicking **Restart**. If the problem persists, shut down and restart the Mellosotron.



## License information

Last updated: October 27, 1998

This file contains important license information regarding the distribution and use of the Mellosoftron.

The Mellosoftron is NOT in the public domain. It is [Copyright \(C\) Tommy Anderberg, 1996-1998](#). All rights reserved worldwide. This software and all the accompanying data and documentation files are protected by national and European laws and by International Treaty provisions. You must treat the Mellosoftron like any other copyrighted material, except that you may copy the Mellosoftron Evaluation Package and give to others (subject to the limitations outlined below). Any use or distribution of this software, data and documentation in violation of Copyright law or of the terms outlined in this document will be prosecuted to the best of our ability.

The conditions under which you may copy and distribute the Mellosoftron are described below under [Limited Distribution License Agreement](#).

The conditions under which you may use the Mellosoftron are described below under [Limited User License Agreement](#).

All rights not expressly granted here are reserved to Tommy Anderberg.

### **Limited Distribution License Agreement**

This Limited Distribution License Agreement (throughout this section, the "Agreement") is a legal agreement between you, the distributor, and Tommy Anderberg. By copying any part of the Mellosoftron, you are agreeing to be bound by the terms of this Agreement.

As the copyright and trademark holder for the Mellosoftron, Tommy Anderberg authorizes individuals and organizations to copy the files contained in the Mellosoftron Evaluation Package for their own use or for giving to others, ONLY when the following conditions are met:

- The Mellosoftron Evaluation Package - including all related program, data and documentation files - may not be modified in any way and must be distributed together, as a complete package, without exception. You may not distribute corrupted or otherwise damaged files.
- No price or other compensation may be charged for the Mellosoftron Evaluation Package. A reasonable distribution cost may be charged for the distribution media, shipping and handling. BBS systems may not charge beyond their normal rates for downloads of the Mellosoftron Evaluation Package.
- The Mellosoftron is designed to be easily extended by the addition of instrument patches and samples. You may not, however, distribute an extended version of the Mellosoftron Evaluation Package without prior written permission from Tommy Anderberg.
- Neither this Agreement nor any part or portion hereof shall be assigned or sublicensed, except as described herein. Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby. If any provision is determined to be

unenforceable, you agree to a modification of such provision to provide for enforcement of the provision's intent, to the extent permitted by applicable law. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any terms of this Agreement, your license is automatically terminated, and you may be prosecuted at any time under national and European laws and under International Treaty provisions.

- You acknowledge that you have read this Agreement, you understand this agreement, and understand that by copying any part of the Mellosotron, you agree to be bound by this Agreement's terms and conditions. You further agree that, except for written separate agreements between Tommy Anderberg and you, this Agreement is a complete and exclusive statement of your distribution rights. This agreement supersedes all prior oral agreements, proposals or understandings, and any other communications between Tommy Anderberg and you relating to the subject matter of this Agreement.

### **Limited User License Agreement**

This Limited User License Agreement (throughout this section, the "Agreement") is a legal agreement between you, the user, and Tommy Anderberg. By installing the Mellosotron, by loading or running any part of it, or by placing or copying any part of it onto your computer hard drive, you are agreeing to be bound by the terms of this Agreement.

For the purposes of this section, "use" means loading the software into RAM, as well as installation on a hard disk or other storage device.

As the copyright and trademark holder for the Mellosotron, Tommy Anderberg authorizes individuals or organizations to use the Mellosotron, ONLY when the following conditions are met:

- You shall not: modify, translate, disassemble, decompile or reverse engineer the compiled executables contained in the Mellosotron.
- You may not exploit running copies of the Mellosotron commercially, e.g. by renting, leasing or selling access to a computer running the Mellosotron.
- You may not distribute or otherwise exploit the output produced by the Mellosotron if you have not purchased your own personal Mellosotron key from Polyhedric Software.
- You are free to distribute and exploit the output produced by the Mellosotron, provided that you have purchased a personal copy of the Mellosotron, and that the output in question is not part of a competing product (e.g. a sound library for use with the Mellosotron, the MIDNight Express or WAVmaker), in which case you must have prior written permission from Tommy Anderberg.
- The user accepts sole and full responsibility for any damage or loss arising from the use or the inability to use the Mellosotron. While this software has been extensively tested and is believed to be safe, Tommy Anderberg and Polyhedric Software disclaim any warranties, either expressed or implied, including but not limited to warranties of fitness for ANY particular purpose and of uninterrupted, error free operation. Under no circumstances shall Tommy Anderberg or Polyhedric Software be liable for loss of data, loss of profits, lost savings, special, incidental, consequential, indirect or other similar damages arising from the use or the inability to use the Mellosotron.

- Neither this Agreement nor any part or portion hereof shall be assigned or sublicensed, except as described herein. Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby. If any provision is determined to be unenforceable, you agree to a modification of such provision to provide for enforcement of the provision's intent, to the extent permitted by applicable law. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any terms of this Agreement, your license is automatically terminated, and you may be prosecuted at any time under national and European laws and under International Treaty provisions.
- You acknowledge that you have read this Agreement, you understand this Agreement, and understand that by installing any part of the Mellosoftron, by loading or running any part of the Mellosoftron, or by placing or copying any part of the Mellosoftron onto your computer hard drive, you agree to be bound by this Agreement's terms and conditions. You further agree that, except for written separate agreements between Tommy Anderberg and you, this Agreement is a complete and exclusive statement of the rights and liabilities of the parties involved in the use of the Mellosoftron. This agreement supersedes all prior oral agreements, proposals or understandings, and any other communications between Tommy Anderberg and you relating to the subject matter of this Agreement.

## Ordering overview

The freely distributable Mellosoftron Evaluation Package lets you try out the program before you decide to purchase a user license.

To permanently unlock the Mellosoftron, you need to get a personal key from [Polyhedric Software](#). The key comes in the form of a small binary file and is delivered either as an e-mail attachment (this is the fastest and least expensive option) or on a standard 3.5 inch diskette (recommended only if you don't have e-mail access or if your e-mail system can't handle binary attachments).

Please refer to the [License Agreement](#) for legal details regarding the use and distribution of the Mellosoftron and Mellosoftron keys.

Polyhedric Software is letting [Kagi](#) (a specialized order processing firm) handle all orders for Mellosoftron keys. Orders can be submitted to Kagi through an online form, by e-mail, by fax or by snail mail. The online form is the fastest route, snail mail the slowest. Kagi accepts cash (all major currencies), checks, money orders, credit cards (VISA, MasterCard, American Express, Discover or Diners Club), First Virtual and invoices. If you need a paper receipt you can have one sent to you.

A Mellosoftron key is only \$29...

...and you can save \$9 by ordering it with the [GSound 22 CD-ROM](#) or as part of the [WAVmaker Plus](#) or [WAVmaker Studio](#) packages!

- To submit your order online, you need a credit card. A secure form is available for browsers supporting SSL (e.g. newer versions of Netscape Navigator and Internet Explorer). Data entered on the secure form is encrypted before it's sent over the Internet.



[Click here to submit your order online.](#)

- To submit your order by e-mail, fax or snail mail, you need to run Kagi's "Register" wizard. It will prompt you for all the necessary information and encode any credit card data for better security.



[Click here to submit your order by e-mail, fax or snail mail.](#)

If Windows Help complains about not being able to find Register.EXE, you can launch it from the Mellosoftron's folder (click the icon labeled "Order").

Please direct any questions about ordering our products to [sales@polyhedric.com](mailto:sales@polyhedric.com).



[Kagi](#)

1442 - A Walnut Street #392

Berkeley, CA 94709-1405

USA

Fax: +1 510 652 6589

<http://www.kagi.com>

[kagi@kagi.com](mailto:kagi@kagi.com)



To order your Mellosotron key on the web, you need a credit card (VISA, MasterCard, American Express, Discover, Diners Club or JCB Card). A secure form is available for browsers supporting SSL (e.g. recent versions of Netscape Navigator and Internet Explorer). Data submitted through the secure form is encrypted before it's sent over the Internet.



[Click here to submit your order on the web.](#)

To order your Mellosotron key by e-mail, fax or snail mail, you need to run [Kagi's](#) "Register" wizard. It will prompt you for all the necessary information and encode any credit card data (not required - most payment methods are supported) for better security.



[Click here to submit your order by e-mail, fax or snail mail.](#)

If Windows Help complains about not being able to find Register.EXE, you can launch it from the Mellosotron's folder (click the icon labeled "Order").



