

# con Timber ePlayer

for Mac computers

Manual V2.0 17/6/2016

# Table of Contents

Table of Contents .....	2
Credits .....	3
License terms .....	3
Introduction .....	6
Compatibility .....	6
Getting started .....	7
Main Window .....	8
Language Selection .....	9
Quality of Samples .....	9
UDP .....	9
Standard Pitch .....	9
CPU Utilization .....	9
Open Windows and Utilities .....	10
Orchestra Editor .....	11
Voice Column .....	11
Column Playing Mode .....	13
Deleting and Moving Voices .....	13
Adding Voices .....	14
Solo/Mute .....	14
MIDI Activity .....	14
Gain .....	15
Output/Pan .....	15
MIDI channel .....	16
VU-Meter .....	16
Program Editor .....	17
Used RAM .....	17
Voice Select .....	18
Program Select .....	18
Edit the Program List and the Names .....	18
Open the Keyboard .....	19
MIDI pitch and dynamic .....	19
Setting the Timbre .....	19
Link a Program to a Playing Mode .....	20
Link a Note .....	21
Interpret Chords .....	23
Ensemble play (Chorus) .....	24
Notation Graphics on/off .....	24
Show Sound Info .....	25
Keyboard .....	27
Key detune, key gain and loop mode .....	28
Decay .....	29
Reverb .....	30
Record .....	30
Load orchestra .....	32
Save orchestra .....	32
Audio Settings .....	33

VU Meter .....	34
MIDI in .....	35
Playing scores .....	35
Example 1: with Finale™ .....	36
Example 2: with Cubase™ .....	37
Legato effect .....	38
Patching audio outputs .....	39
MAX and OSC messages .....	39
Glissando commands .....	43
Dynamic commands .....	44
Reverb commands .....	46
Decay command .....	47
Miscellaneous.....	48
Troubleshooting .....	49
Glossary .....	50

## Credits

Thomas Hummel, concept, software, recordings. Sven Hinz, cutter. Mike Murphy, software. Julius Holderer, software. Daniel Figols, Mark Barden, translations. Jo Schlosser, recordings. Roberto Fabbriciani, flute. Martin Fahlenbock, flute. Antje Thierbach, oboe. Christian Hommel, oboe. Sascha Armbruster, saxophone. Rico Gubler, saxophone. Matthias Stich, saxophone. Ingolfur Vilhjalmsson, clarinet. Andrea Nagy, clarinet. Nicola Miorada, clarinet. Johannes Schwarz and Annette Winker, bassoon. Steve Altoft, trumpets. Delphine Gauthier-Guiche, horn. Thomas Wagner, bass trumpet, trombone, tuba. Jörgen Welander, tuba. Klaus Steffes-Holländer, piano. Ursula Eisert, harp. Christian Dierstein, percussion. Konrad Graf, timpani. Pascal Pons, percussion. Yuriko Sekiguchi, percussion. Peter Kleindienst, guitar. Alexander Lehner, guitar. Annette Rießner, accordion. Stefan Häussler, violin. Thomas Hofer, violin. Patrick Jüdt, viola. Beverley Ellis, violoncello. Johannes Nied, double bass.

ePlayer is a term suggested by the philosopher Harry Lehmann.

# License terms&Conditions

The licensing terms described herein – in the version applicable on the date of the order – pertain to contracts on the purchase of software products (software and databases) between conTimbre and its clients. Also applicable in addition to these special licensing terms for software products are the Terms and Conditions of conTimbre.

## A. Software

In particular, the subject of these licensing terms is the granting of a license for conTimbre software. The provisions of this licensing agreement also apply to the right to use updates, extensions, add-on components, or other additional programs intended for use with the software that are provided to the client by conTimbre.

### 1. Licensing

conTimbre grants the client a simple, non-exclusive, and temporally as well as spatially unrestricted license to use the software in machine-readable form (object code) as well as the documentation in conformity with the contract. The documentation meant in this context is integrated electronically into the software or stored on the provided data storage device. The client shall not be granted any further rights to the software and the documentation.

### 2. Single- and multi-user licenses

2.1 The client shall only be entitled to install, use, access, possess, and run a copy of the software on a single computer. The software shall not be used over a network unless the client has explicitly been granted a network license.

2.2 If the software is to be made available for use over a local-area network, the client must purchase a separate network license. If the network license is granted for a single user (single-user license), only a single person may use the software at the same time. If the network license is granted for more than one user (multi-user license), only the amount of persons for which the license was granted shall be authorized to use the software at the same time.

2.3 The network license shall not authorize the client to use the software in a public network or make it available to the public in any other form. If the software is to be used over WLAN, the client must take adequate data encryption measures to prevent third persons from accessing the licensed material.

### 3. Limits to the License

3.1. The license is limited to the object code of the software. conTimbre is not obliged to make the source code of the software available to the client. The client is prohibited from reverse engineering or reassembling the object code of the software or otherwise exposing, editing, or changing it. The client is only authorized to decompile the object code according to legal regulations, in particular only if it is necessary to establish interoperability with other programs, if the necessary information has not been made available to the client, and if the decompilation is limited to the parts of the original software.

3.2 The client may only make one backup copy, exclusively for archiving purposes and for personal use. Any further reproduction of the software shall not be permitted unless it is necessary for the proper execution of the software (e.g., for display on a monitor and as intermediate storage in RAM).

3.3 The software may only be used as an integrated whole. Separating components for individual use shall not be permitted.

3.4 If the software provided to the client includes documentation that is integrated electronically into the software or stored on the provided data storage device (electronic documentation), the client shall be authorized to make a printout of this electronic documentation. Any further reproduction of the documentation not necessary for displaying it on the computer monitor, editing the documentation, or making it available to the public shall not be permitted.

#### 4. Transfer

4.1 The license shall not be shared. The client is only authorized to transfer the rights described in this license agreement if he or she adheres to the following conditions:

- a) the terms of the license described herein shall be transferred in their entirety;
- b) the client shall transfer all copies of the software to the new licensee, including the data storage devices and documentation belonging to them;
- c) the client shall delete the installed software so that recovery is impossible; and
- d) the new licensee shall agree to these licensing terms.

4.2 Leasing or otherwise allowing a third party to use the software in return for payment is not permitted. A sub-licensing is not permitted.

4.3. If the customer purchases a student license or an academic license, the customer has the above mentioned rights only under the condition that he does not distribute the software and the documentation or transfer, sell, lease, change, license or transfer it in another way.

#### **B. Violation of Licensing Terms**

If the client violates these licensing terms, conTimbre is entitled to withdraw the license for the software and the documentation or the database, unless the client was not responsible for the violation. Other legal rights remain unaffected. In particular, the client is advised that conTimbre is authorized to assert claims pursuant to §§ 97 et seq. of the German Copyright Act (UrhG) (e.g., compensation for damages, restitution of profits gained, removal, et al.).

#### **C. Exclusion of Liability**

Except in the case of death, bodily injury, or illness, conTimbre shall only be liable for damages incurred through deliberate or grossly negligent action or the intentional breach of an essential contractual obligation on the part of conTimbre, its employees, or its vicarious agents. This shall also be true of damages resulting from the breach of obligations in contract negotiations as well as the performance of tortious acts. Any further claims for damages shall be excluded.

The preceding limitations of liability shall not apply in the case of death, bodily injury, or illness, for a defect covered by a guarantee that has been given for the condition of the product, and in the case of fraudulent concealment of a defect. Liability in accordance with the Product Liability Act remains unaffected.

#### **D. Applicable Law**

The licensing terms described herein are subject to the law of the Federal Republic of Germany to the exclusion of the United Nations Convention on Contracts for the International Sale of Goods and private international law.

# Introduction

The ePlayer is a sampler. It comprises of a database with more than 150 Instruments, sorted by their group and family. More than 4,000 playing techniques are available, from simple ordinary to sophisticated new music techniques. For certain playing techniques such as multiphonics, chords may be interpreted as one sound. It offers the possibility to display in real time additional information, photos and graphics about the currently played sound. The ePlayer is compatible with Open Sound Control (OSC) messages, and is available in both German and English languages.

The ePlayer may be played both from a MIDI keyboard, and from score editors (like NoteAbility™ (Mac only), Sibelius™, Finale™).

The score editors may play through the MIDI IO using a MIDI interface.

On Mac: You may use also IAC busses if you want to use the ePlayer together with score editors etc.

On Windows-Computers: You may also use third party software like LoopBe and other third party software if you want to use the ePlayer together with score editors etc.

Using OSC, you have more possibilities, such as microtonal pitches and automatic creation of voices. The most comprehensive control is offered if the ePlayer is used within a MAX environment.

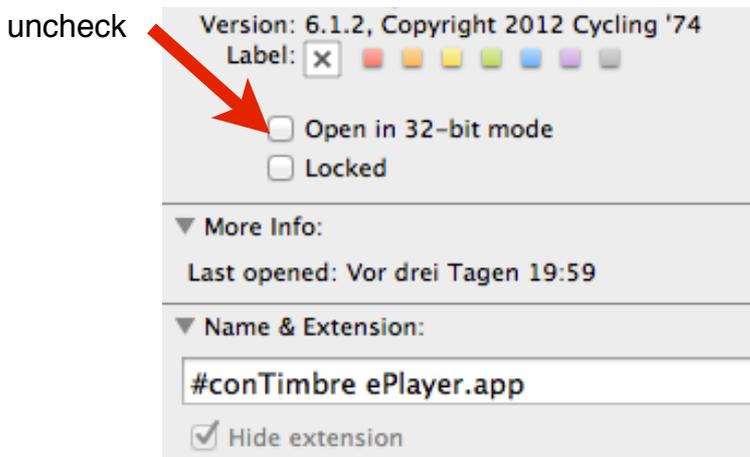
# Compatibility

The Macintosh ePlayer standalone application runs on

- Apple computers running MacOS 10.7.or newer.
- PC-Computers running windows 7 or newer.
- Minimum RAM 4GB.

The ePlayer maxmsp patch (ePlayer\_maxmsp/#contimbre ePlayer) needs MAX7. ([www.cycling74.com](http://www.cycling74.com)) runtime version or full version installed.

- On Mac: Uncheck „open in 32 bit modus“ in the info window of the Max program file.



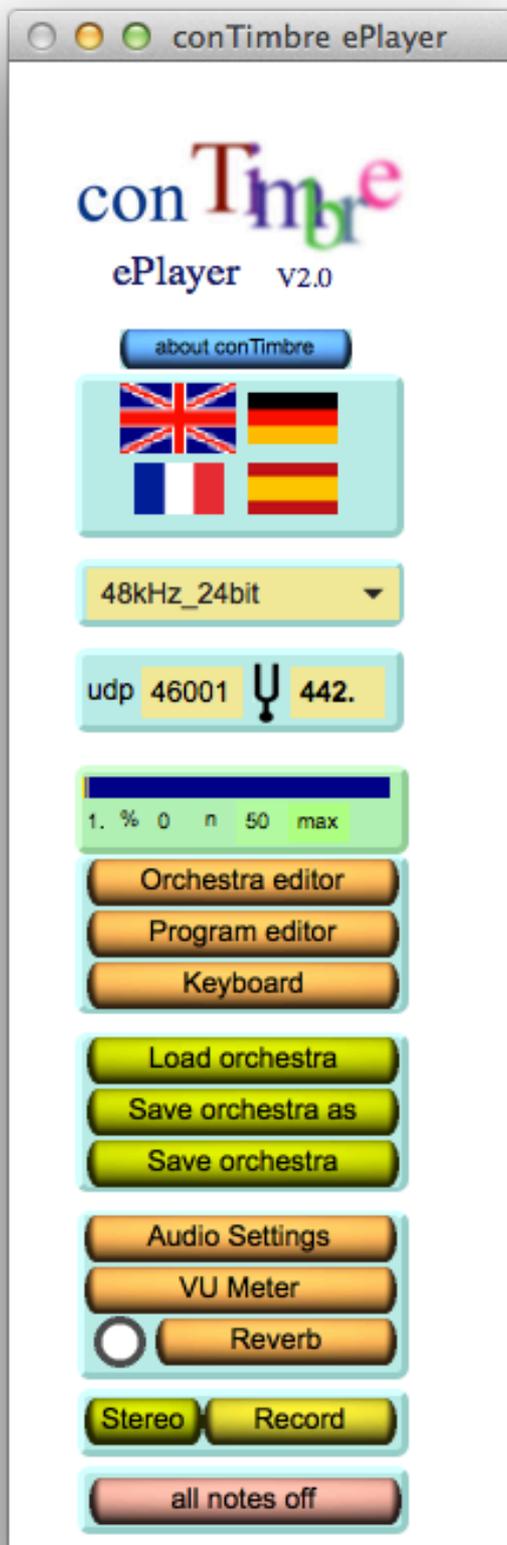
Generally, the stability of the ePlayer maxmsp version depends on modifications and additions by the user and eventually the exact version of MAX.

# Getting started

- Make a copy of the complete database onto your computer or your external drive. Do not place the conTimbre database in higher folder hierarchies as the software might not find files with long names.
- Insert the conTimbre USB stick.
- Start
  - i) Start *ePlayer\_standalone/#contimbre ePlayer*
  - ii) Start *MAX/MAX-runtime* and then load *#conTimbre ePlayer* in folder *ePlayer\_maxmsp*. Be careful not to load more than one instance of *#conTimbre ePlayer.maxpat*.

**Important:** For proper functionality, the conTimbre USB stick must stay inserted while you are using the ePlayer.

# Main window



## Language selection



Click to select your preferred language.

## Quality of samples

48kHz\_24bit

The first value indicates the sample rate in kHz of the samples used. The second value indicates the bit depth of the samples used. Higher values have improved quality, but use more RAM.

## UDP

udp 46001

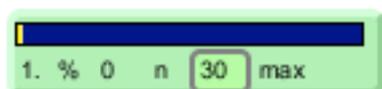
If you want to use OSC-messages to control the ePlayer, you have to set the UDP port ePlayer will listen on. The default is 3000.

## Standard Pitch

U 400.

You can tune the ePlayer to a certain standard pitch. The default is 442. Hz.

## CPU Utilization



percentage of CPU taken  
current number of notes playing  
maximal number of notes played at same time

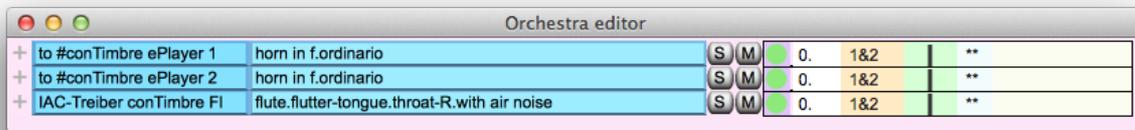
Maximum number of notes played at same time:

This number is set to avoid audio clicks resulting from high CPU utilisation or strongly delayed notes, depending on the audio settings. If you try to play more notes than this limit, the oldest playing note is switched off before the new note is played (note stealing). In this case, the new note may start with a small delay.

## Other menu items

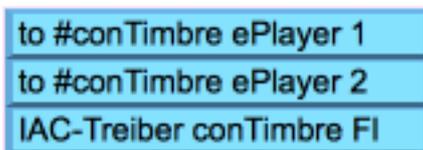
	opens the orchestra editor.
	opens the program editor.
	shows the keyboard.
	loads an orchestra.
	saves the orchestra again with the last name used. Or press ctrl-s.
	saves an orchestra with a new name.
	lets you edit the audio settings in your environment.
	opens the VU Meter window.
	switches reverb on and opens reverb window.
	records sound files.
	
	switches all notes off.

# Orchestra Editor



Each line in the orchestra editor window corresponds to a voice. On startup of the ePlayer, the voices are created just from the MIDI IO setup. Each voice can receive messages only from one MIDI IO device.

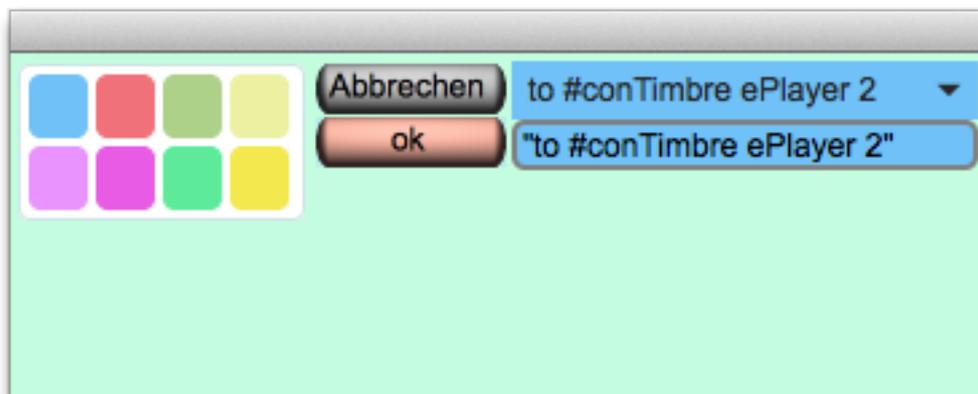
## Voice Column



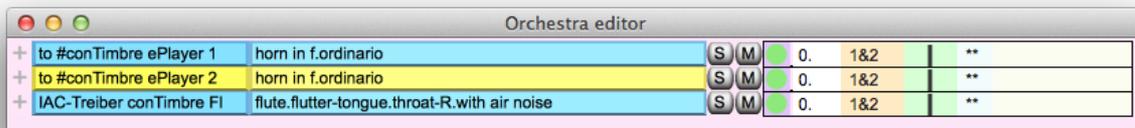
This column represents the names of the voices. If the name starts with the name of a MIDI device, it will react to input from this MIDI device. The name is written in black letters.

If the new name does not start with the name of a MIDI device, it will be displayed in red letters within the orchestra editor. In this case, the voice will only react to appropriate MAX messages or OSC messages (see the chapter "MAX and OSC messages").

When clicking on a line, you may edit the color of the line or change the voice name:



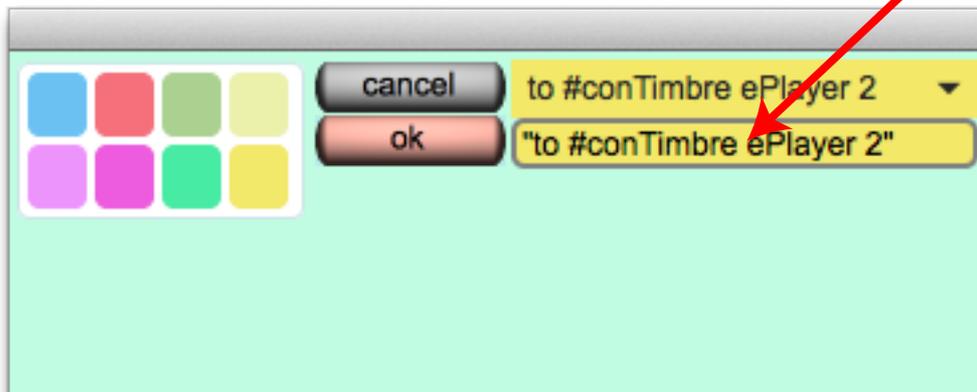
You can click on a color to color the voice differently:



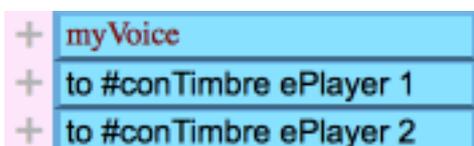
With the popup menu, you can select another MIDI input for this voice:



A text field allows you to set another name:



If the new name does not start with the name of a MIDI device, it will be displayed in red letters within the orchestra editor. In this case, the voice will only react to appropriate MAX messages or OSC messages (see the chapter "MAX and OSC messages"):



If the new name starts with the name of a MIDI device, it will react to input from this MIDI device. It is therefore possible to play several voices with one MIDI device. In the following example, all voices will react on MIDI input from the device "IAC-Treiber conTimbre".

IAC-Treiber conTimbreFl
IAC-Treiber conTimbreVI
IAC-Treiber conTimbreVc

### Column Playing Mode

horn in f.ordinario
horn in f.ordinario
flute.flutter-tongue.throat-R.with air noise

This column shows the names of the current programs of all voices. When you click on a program name, the program editor window opens for this voice and this program.

### Deleting and Moving Voices

Use the Shift-key or the Command-key and click into the voice names area in order to select some voices (or use the ESC-key to cancel).

+	IAC-Treiber conTimbre1	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre Vc	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre db	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre Fl	flute.flutter-tongue.throat-R.with air noise	S	M

Now, you may click onto the selected voices and drag them to the desired position:

+	IAC-Treiber conTimbre1	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre Vc	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre db	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre Fl	flute.flutter-tongue.throat-R.with air noise	S	M



The result in this example would be

+	IAC-Treiber conTimbre1	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre db	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre Fl	flute.flutter-tongue.throat-R.with air noise	S	M
+	IAC-Treiber conTimbre	horn in f.ordinario	S	M
+	IAC-Treiber conTimbre Vc	flute.flutter-tongue.throat-R.with air noise	S	M

Too, you may select voices and delete them with  
 - the Backspace key (Mac)  
 - the delete key (Windows-PC).

**Warning:** You cannot undo the delete.

### Adding Voices



+	IAC-Treiber conTimbre1	
+	IAC-Treiber conTimbre db	
+	IAC-Treiber conTimbre Fl	
+	IAC-Treiber conTimbre	
+	IAC-Treiber conTimbre Vc	

With the +-button, you can duplicate the corresponding voice. The maximum number of voices is 128. On windows, an interface for 128 voices maximum and 32 voices maximum (faster loading) is available.

### Solo/mute

You may solo/mute voices:

solo

mute



S	M	0.	1&2	1
S	M	0.	1	2
S	M	0.	2	3
S	M	0.	5&6	**
S	M	-5.	5&6	**

## MIDI Activity

S	M	0.	1&2	1
S	M	0.	1	2
S	M	0.	2	3
S	M	0.	5&6	**
S	M	-5.	5&6	**

These buttons are highlighted when a MIDI/OSC message was received.

## Gain

S	M	0.	1&2	1
S	M	0.	1	2
S	M	0.	2	3
S	M	0.	5&6	**
S	M	-5.	5&6	**

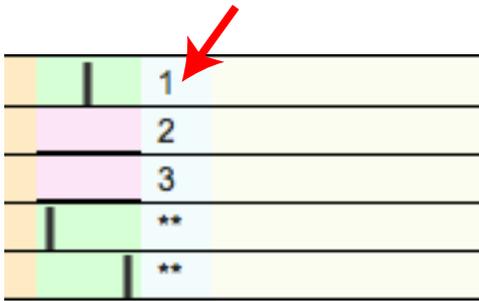
Here you can change gains of voices in dB. Any sound played on this voice will be softer or louder. However, conTimbre ePlayer plays sounds a priori in their natural loudness balance. If you change the gain, it is no longer the natural balance.

## Output/Pan

Output	Pan
1&2	1
1	2
2	3
5&6	**
5&6	**

You can individually assign mono or stereo outputs for each voice. The voice will then play through the specified channel(s) of your audio interface. When selecting stereo outputs, you may set a left/right pan.

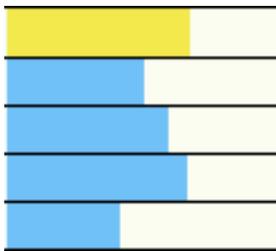
## MIDI Channel



1
2
3
**
**

For each voice, you can set an individual MIDI channel. This option is only available if the voice starts with the name of a MIDI device (the voice name is displayed in black letters). \*\* means that the voice accepts all MIDI channels.

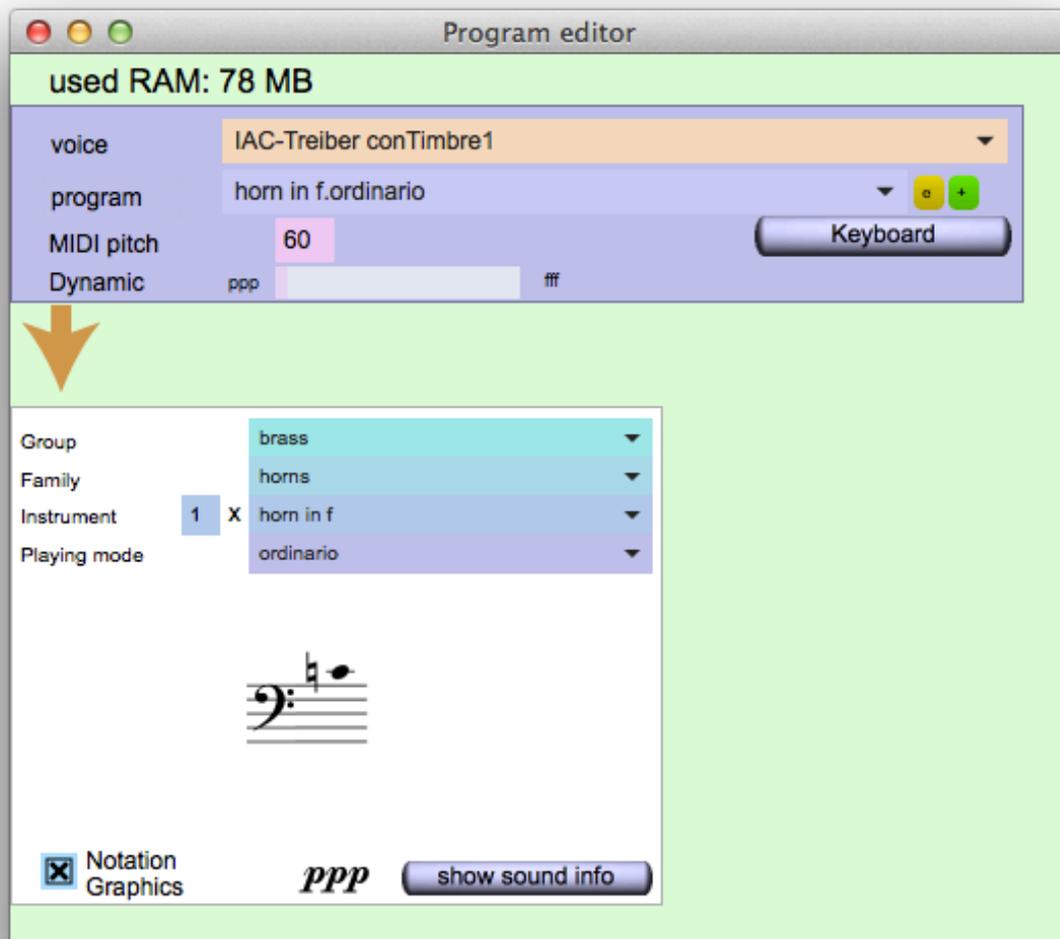
## VU-Meter



The loudness in each audio output is shown in the VU meter column.

# Program Editor

When clicking the program editor button in the main window or into the program name column of the orchestra editor, the program editor window opens.



## Used RAM

At the top, you see always the RAM amount which is already taken by samples:

used RAM: 10 MB

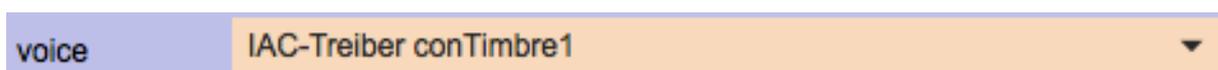
In the conTimbre ePlayer, necessary samples are always preloaded into RAM. There is no real time reading of sound files from disk.

On Mac-Computers: Check your RAM using the Apple Activity Monitor Application/Option System Memory to know how many samples you can load.  
On Windows-Computers: Check your RAM using the Task Manager to know how many samples you can load.

The ePlayer gets less reliable when no more free RAM is available ie. the Virtual Memory is used.

## Voice Select

You can select a voice for which you want to edit programs:



## Program select

You can select the program to be edited:



The program name may be any arbitrary name and is just set by default. A program name does not necessarily correspond to the content of the program. If the name of the program starts with a number and a space, the program may be recalled by MIDI program change messages with the corresponding number. When a MIDI program change message is received for this voice, the program with the appropriate number will be called, e.g.:



This program may be recalled with the MIDI program change command 1.

## Edit the Program List and the Names

• Rename the program: press the -button, type in the new name of the program and press "return"-key:



• Add a new program: press the -button. By default, it is just a copy of the current program.

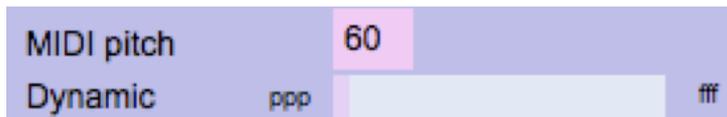
• Delete this program or all programs but the first one: press the -button. The last remaining program cannot be deleted.

## Open the keyboard

When clicking , the keyboard layout of the current program opens. It shows details of the selected program.

## MIDI Pitch and Dynamic

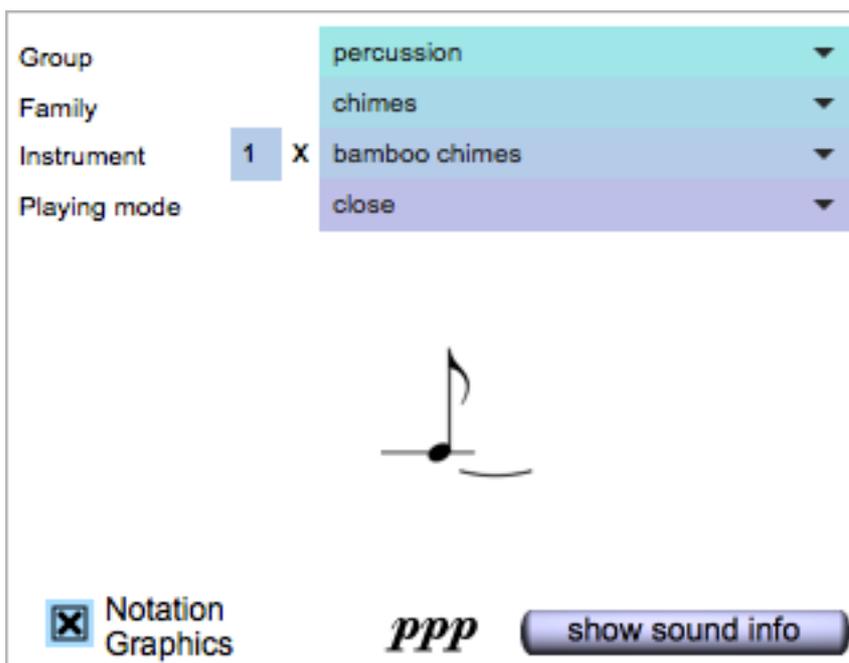
The MIDI pitch and the dynamic of the last note played or the note to be edited is shown.



You can set the pitch.

## Setting the Timbre

The lower part of the "edit programs"-window presents the timbre setting of the current pitch or the current program, respectively.



The timbre setting of the program or of a single pitch may be edited here.

There are three possibilities:

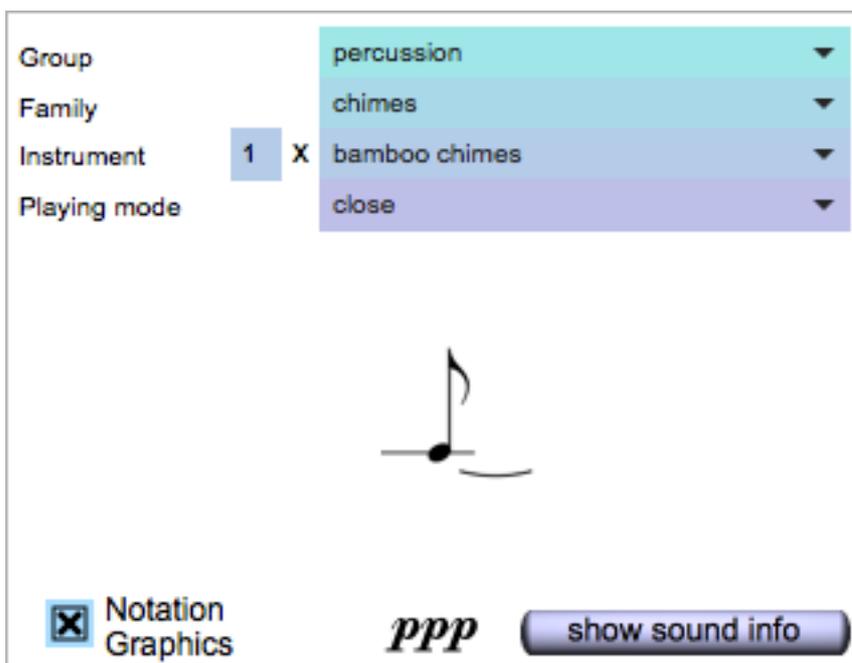
- link the complete keyboard to a certain playing mode.
- link one pitch to a certain sound.
- interpret chords.

## Link a Program to a Playing Mode

First define the current program shown in the upper part of the window, e.g.



to be a playing mode presented in the lower part:



In the dialog being presented, select "link mode":



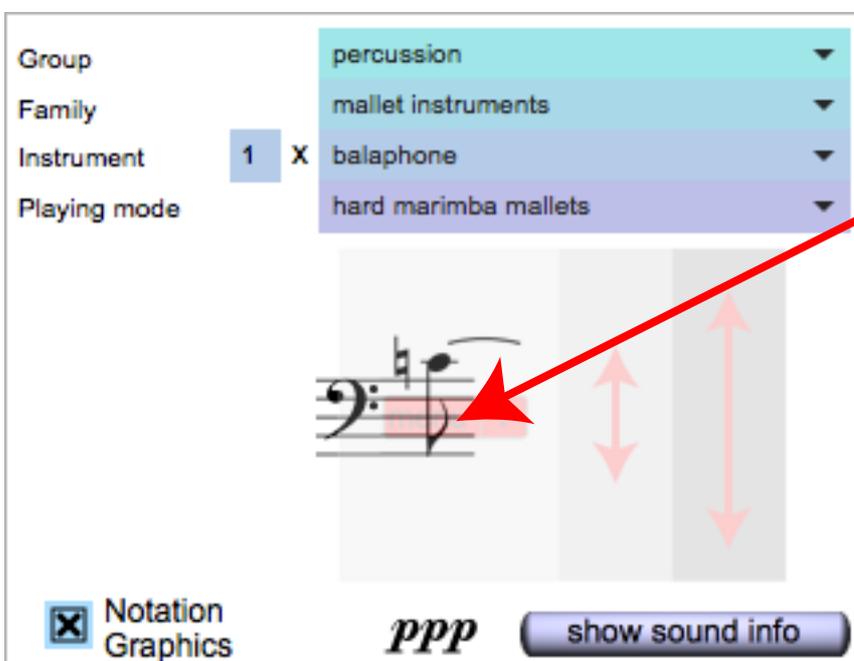
+xxxMB indicates how much RAM this link would need additionally to the RAM already taken. For some playing techniques, e.g. woodwind multiphonic lists, the amount of available sound versions exceeds the size of the keyboard. In this case, only some of the versions will be loaded. However, it is possible to edit the choice of sound versions with the option "link note" and select from the complete set of sound versions.

## Link a Note

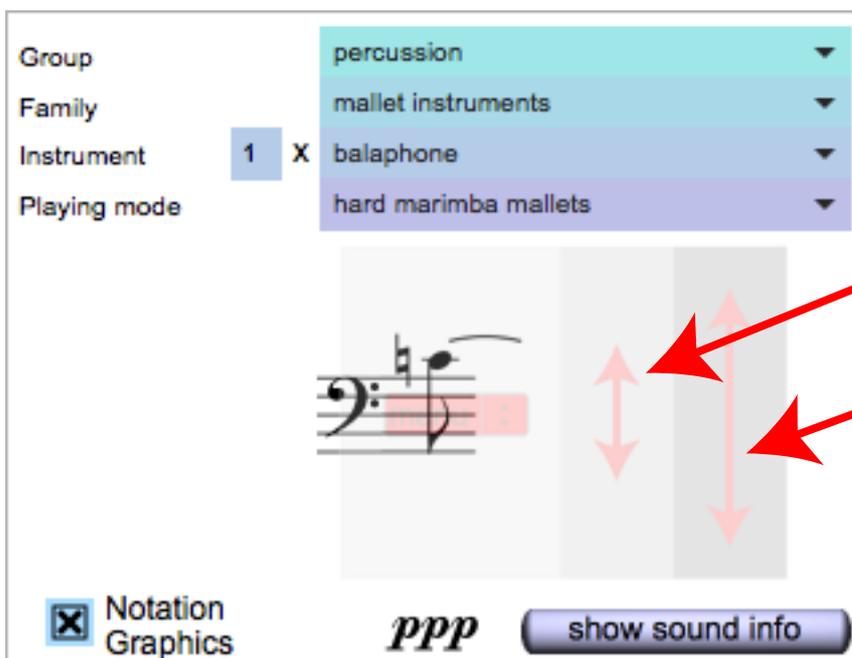
If you only want to define a sound for the current pitch in the current program, you may first set the desired pitch in the upper part of the program editor:



In the lower part, you may set the group, family, instrument and playing mode as described in the option "link mode". Furthermore, you can select the desired sound. If you click in the left part of the note region, you can select the note with a popup menu.



In the half right part, you can click and drag vertically to select a note.  
In the very right part, same is possible with higher speed.



In the dialog, select "link note":



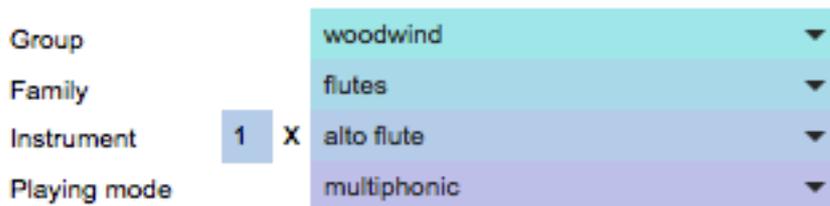
+xxxMB indicates how much RAM this link would need additionally to the RAM already taken. All other pitches of the keyboard layout are not affected.

## Interpret Chords

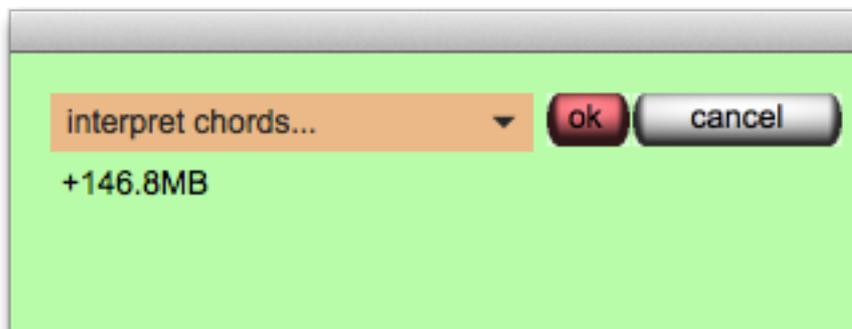
For some playing techniques like woodwind multiphonics, play & sing, and harmonics, a special chord mode may be selected.  
E.g. the playing mode



may be linked to the following playing mode:



In the dialog, select "interpret chords":



+xxxMB indicates how much RAM this link would need additionally to the RAM already taken.

In the "interpret chords" mode, chords with several MIDI-pitches are interpreted as one sound. The ePlayer will find the sound with a partial structure which fits best to the played chord. If several sound match with the same similarity, the ePlayer rows them up with different velocity zones according to their amplitude. This means, playing the chord softly will result in a soft sound and playing the chord loudly, a loud sound of similar partial structure will be played.

The "interpret chords" mode is possible for:

- multiphonics. If a chord is played, the ePlayer searches in real time for a multiphonic whose partials fit to the chord. Only nonharmonic multiphonics are taken into account.

- play and sing. A chord with two notes will be interpreted so that the first note is the played note and the second is the note sung.



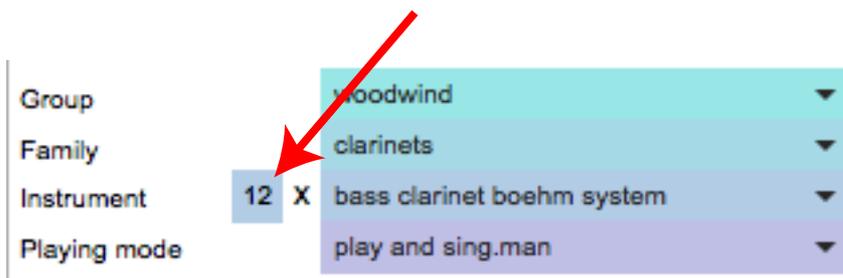
- harmonics. The lower note is interpreted as the fundamental, the higher as the partial node:



- some whistle tones of woodwind instruments are treated like harmonics.

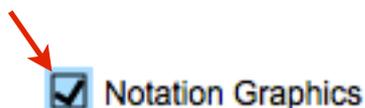
### Ensemble play (Chorus)

You can set the number of instruments playing a voice, creating an ensemble. This is also known as the chorus effect. Please note, the number of instruments will correspond to the number of notes triggered with a single noteon.



### Notation Graphics on/off

The time precision during playback of scores may be lower, when notation graphics and sound info graphics are continuously updated. Suppress notation graphic update here:



## Show Sound Info

**show sound info** opens a menu which displays more information about the last selected or played sound. The menu works in real time. It is possible to use it while playing.

### sound info

Instrument

file name: contrabass trombone.m52.F2.Wnc1568

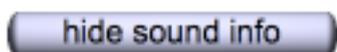
contrabass trombone with mouthpiece of baritone saxophone



In the popup menu, you may select different types of information:

- Option Instrument. The instrument and eventual accessories or describing photos are presented.
- Option Comments
- Option Partial.
- Option Spectrum. The spectral envelope and the spectral center is displayed. The spectral center is musically equivalent to the brilliance of a sound. A sound with a high spectral center sounds clear, a sound with low center sounds dull.
- Option Other. It is displayed:
  - the absolute intensity. The absolute intensity of a sound is the maximal intensity in dBA at a distance of 1m.
  - the attack time.
  - the spectral complexity. A sound with low spectral complexity (close to zero) resembles a sine wave. A sound with high complexity (close to 1) resembles white noise.

You may hide the information with this button:



All pitch, partials and intensity informations hold true for the corresponding sample. The file name of the sample is indicated underneath the popup menu. As the ePlayer transposes sounds and interpolates their intensities, deviations may occur in relation to the played sound.

# Keyboard

The window keyboard shows details of programs and allows to play test tones. At the top, you may select the voice and the program.



In the main section a keyboard with 128 pitches is presented.

The left scroll bar helps you to scroll through the keyboard.

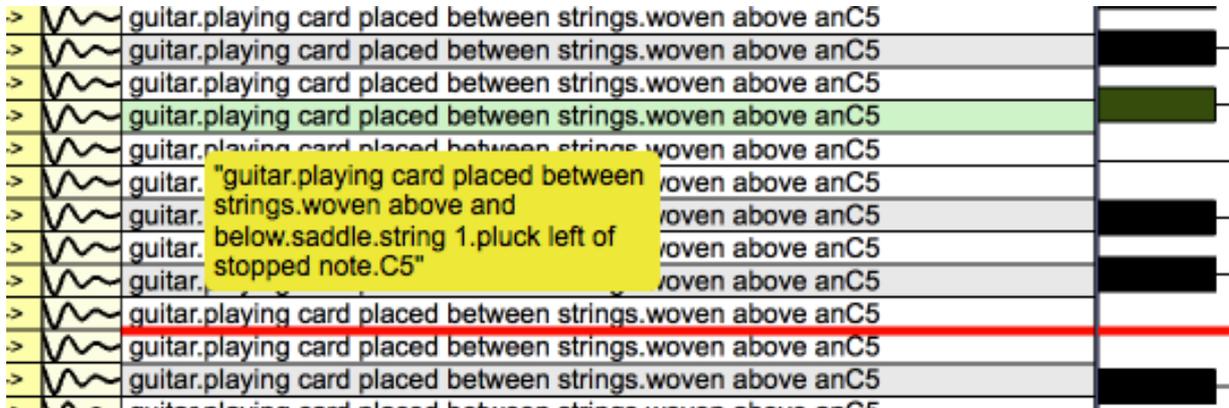


The horizontal red line indicates middle c.

You may click the keyboard to play short test tones.

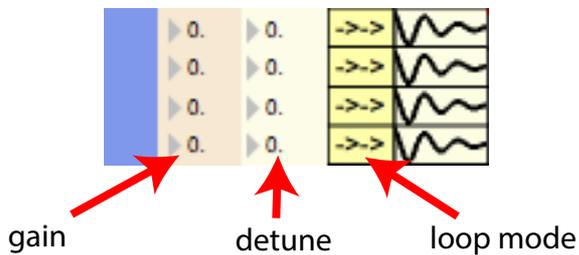
The sample names of each pitch are presented. If the displayed description name of a sample is a shortcut, you may place the mouse over a key.

The full description name of the sample will be displayed:

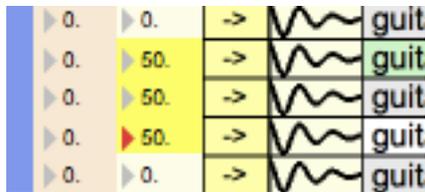


## Key detune, key gain and loop mode

At the left side of the keyboard, you may change the gain of single keys (in dB), detune single keys (in cents) or change the loop mode.



Click into a number box and type in a new value or drag the mouse while clicking. When a number box is clicked (red triangle), you may copy this value into adjacent keys just by using the up or down key:



The key detune option is mainly included for microtonal MIDI keyboard playing. However, it applies for all MAX and OSC commands.

Click into the loop mode option, to switch between different types of loop modes for one key. With alt-click or ctrl-click, you may switch all keys at same time.

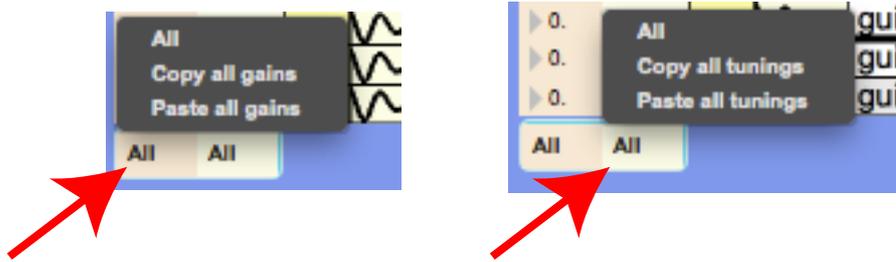
You can select between 3 loop modes:

- no loop (-> Just play once, then stop)
- normal loop (->->)
- loop forwards and backwards (<->. Sound is first played forwards, than backwards, than again forwards a.s.o.)

For percussive sounds, only the option -> (play once) is offered. The <-> option is only offered if it makes acoustically sense. For this reason, be aware that alt-clicking the loop mode option may not change all loop types in a mixed keyboard setting.

Tunings and gain changes are stored together with the program. Program changes may therefore also be used for the change of the keyboard tuning or key gains.

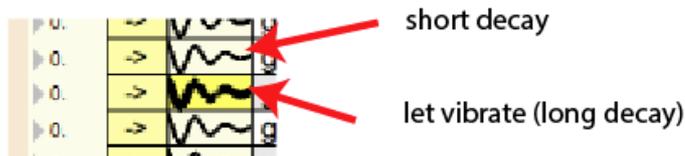
You may copy keyboard tunings and key gains from one program to another using the menus at the bottom of the keyboard window:



1. select the source voice/program
2. press copy...,
3. select the target voice/program
4. press paste....

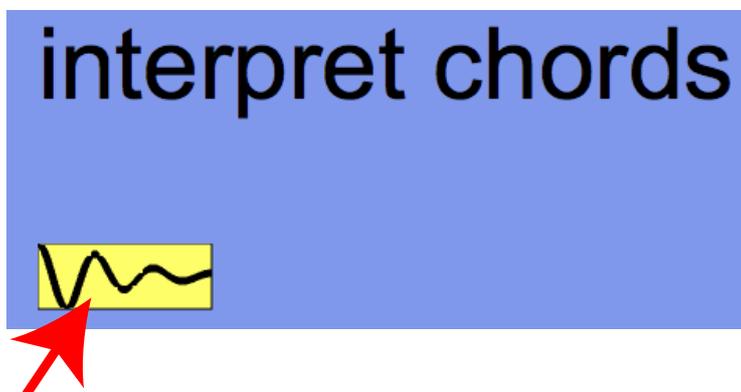
## Decay

If sustained notes are stopped by a noteoff command, you may select the decay mode. The first possibility is a rapid decay which stops the note after at most 100ms. The second possibility, the full decay, may last much longer. By clicking on the  symbol you may switch between the two possibilities for each key of the keyboard. If you click while pressing the alt key, the mode will change for all keys.



A full decay may be stopped immediately by a MIDI *sustain pedal up* command or by the appropriate MAX command.

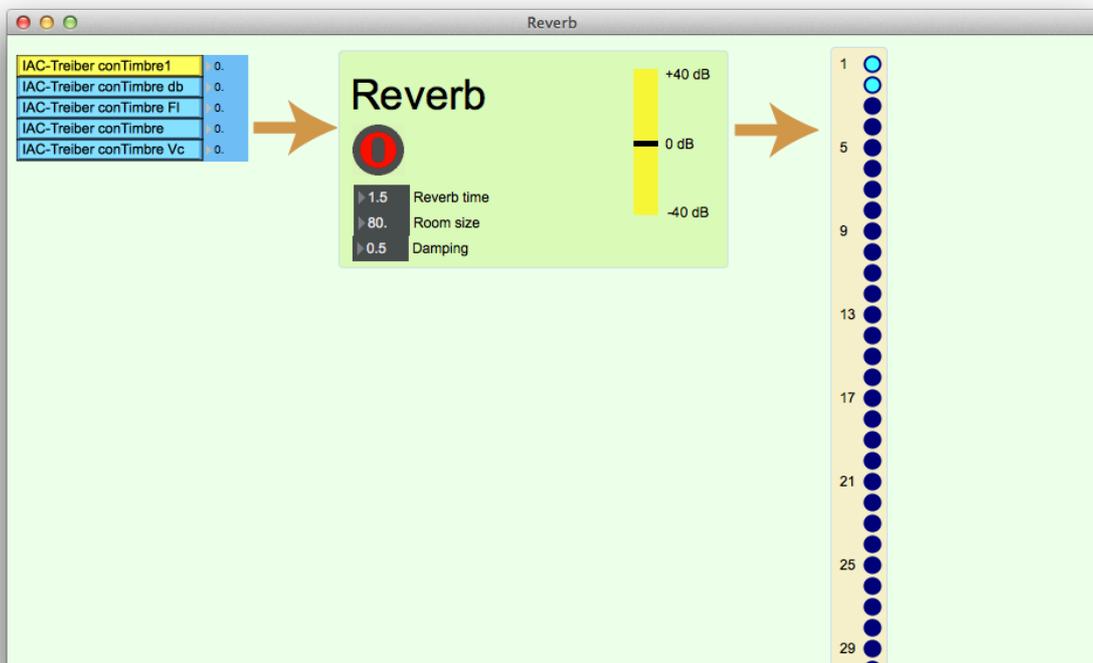
If a program is in the mode „interpret chords“, the decay can only be switched on/off for the complete program.



Let vibrate on/off

# Reverb

The output of the ePlayer may be reverberated. You may switch the reverb on/off:



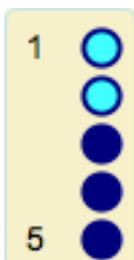
You may set the relative dynamics in dB of the voices into the reverb:



In the central field of the reverb editor the reverb may also be switched on/off. Additionally, the reverb time in seconds, the room size, the low pass damping (0-0.999) and the reverb gain may be set.



In the right part of the window you may select audio output channels for the reverb. In this example, the reverb goes out audio channel 1 and 2:



All settings of the reverb are stored as part of the orchestra.

## Record



Channel Select Button

Record Button

With the record buttons, you may record the ePlayer output to a 24bit high quality sound file.

First, click on the Channel Select button to select the number of audio channels you wish to record. The following possibilities exist:

- Mono (output channel 1)
- Stereo (output channel 1 and 2)
- 8 channels (output channel 1 through 8 in one multichannel file).
- 32 channels (output channel 1 through 32 in 4 8-channel files).

Secondly, click on the Record button. You will be asked for a sound file name. In the case of 32 channel recordings, four 8-channel sound files with the following names will be created:

soundfilename1  
soundfilename2  
soundfilename3  
soundfilename4

After selecting a sound file name, the recorder is ready and blinks:



Clicking on „Start Record“ will start the recording. The elapsed time in hours:minutes:seconds.milliseconds is indicated:



Clicking again the record button/elapsed time button will stop the recording.

## Load Orchestra

Loads all information about voices, their programs, gains, their outputs as well as the overall gain.

On Mac-Computers: Check your RAM using the Apple Activity Monitor

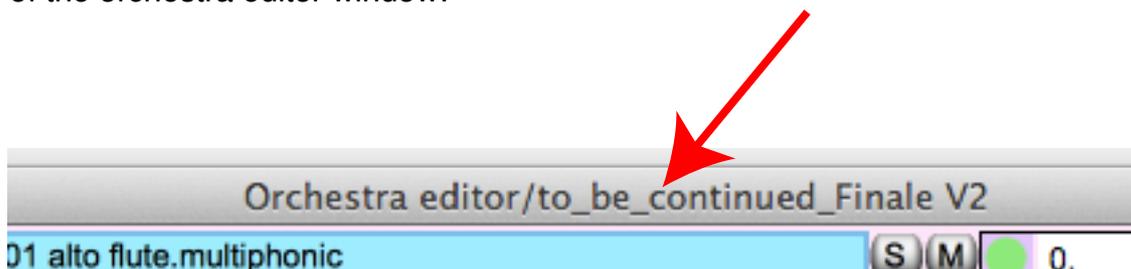
Application/Option System Memory to know how many samples you can load.

On Windows-Computers: Check your RAM using the Task Manager  
to know how many samples you can load.

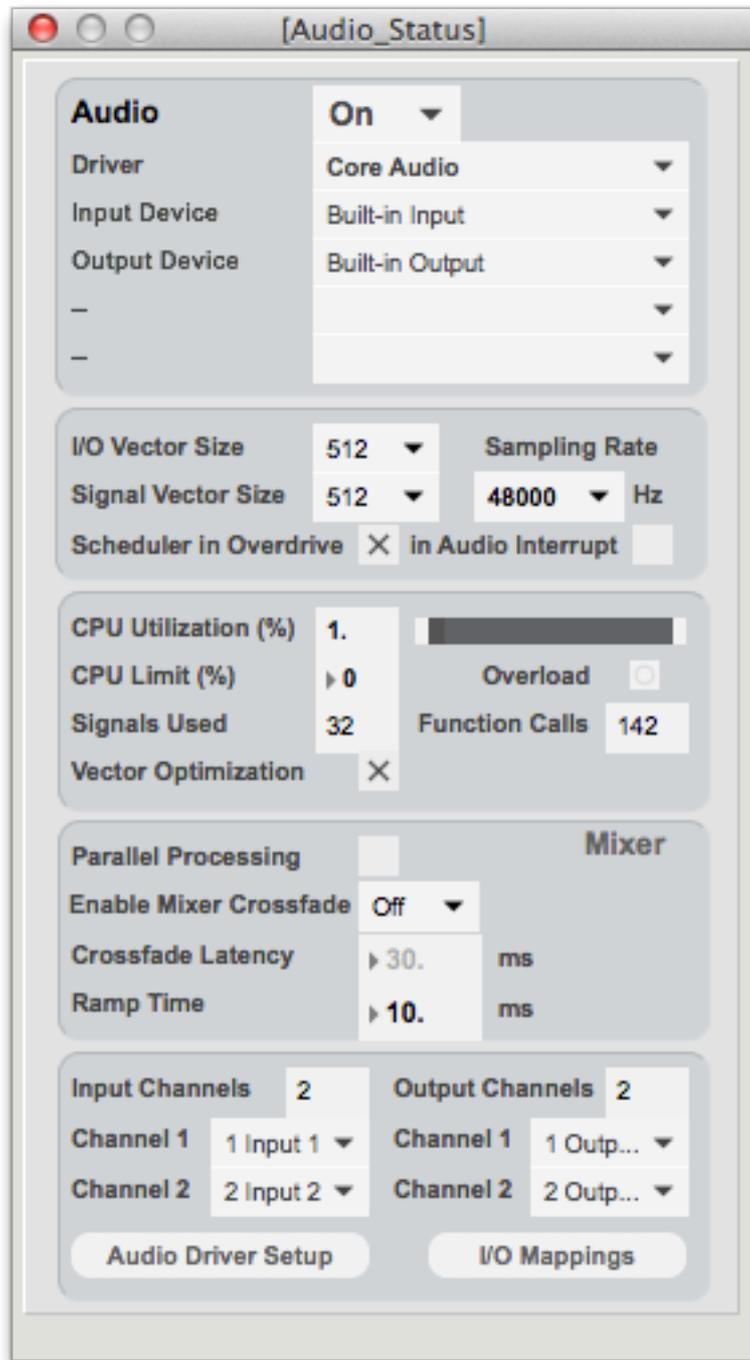
The ePlayer gets less reliable when no more free RAM is available (green color in the Activity Monitor) ie. the Virtual Memory is used. Eventually, lower the samples sample rate in order to load more samples.

## Save Orchestra

Saves an orchestra, including all information about voices, their programs, gains, their outputs as well as the overall gain. The current of the orchestra is displayed in the title of the orchestra editor window:



# Audio Settings



This window is also described in the MAX manual ([www.cycling74.com](http://www.cycling74.com)).

## Driver

Select your preferred audio interface.

## I/O vector size, Signal Vector size:

On windows: ASIO drivers are recommended

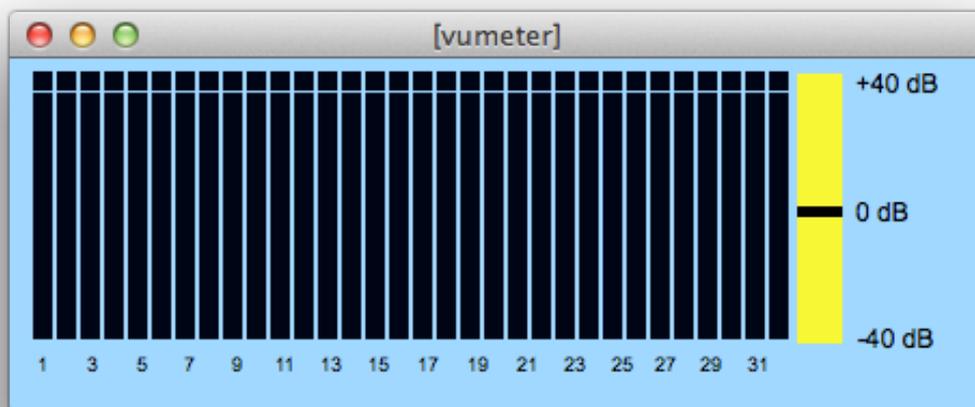
Small I/O vector sizes and small Signal vector sizes will decrease the time latency and increase the CPU utilization.

## Sampling rate:

High sampling rates increase the sound quality of the sample playback, when high resolution

- Scheduler in Overdrive:** samples were loaded. High sampling rates increase the CPU utilization. When switched on, the the latency of notes played is lowered. However, the graphics will update slower. It is recommended to switch on "Scheduler in Overdrive".
- Scheduler in Audio Interrupt:** When switched on, the the latency of notes is lowered. On the other hand, chords with high number of notes may provoke clicks. It is recommended to switch on "Scheduler in Audio Interrupt".
- CPU utilization:** Try to keep the CPU utilisation as low as possible. High CPU utilization may provoke clicks in sound.
- Vector Optimization:** Should always be on.

## VU Meter



The grey bars indicate the VU meter for each audio channel (not voice number!). The slider on the right side corrects the overall loudness.

# MIDI In

If a voice starts with the name of a MIDI device, it will receive MIDI noteon and noteoff messages, program change messages sustain controllers, pitch bend messages and volume messages.

## Note on MIDI messages

If MIDI volume values were sent before, the velocity of the noteon will be scaled down to the last MIDI volume value.

**Pitch bend messages** will be interpreted as follows:

0: transposition minus 200 cents.

64: no transposition

127: transposition plus 197 cents.

A MIDI **program change message** to a certain voice will change to the program of this voice with a name starting with the number of the MIDI program number (afterwards space). E.g., if a program has the name "017 my program", it will be called with a MIDI program change message 17.

**Volume MIDI messages** (controller 7) always try to simulate true crescendi or dimuendi. The timbre of the sound will therefore change within the dynamic change. The algorithm for the interpolation depends on the kind of sound.

# Playing Scores

The ePlayer can play scores from score editors like Finale™, Cubase™, NoteAbility™ and all other score editors capable of MIDI. See the video tutorials at [www.contimbre.com /examples /video](http://www.contimbre.com/examples/video). Proceed as follows:

- Quit the *ePlayer* and your score editor.

## Audio-MIDI-Setup

- Start the Mac application Audio-MIDI-Setup and open the MIDI window.
- Double click on the icon IAC.
- Check *Device is ready*.
- Add a port and give it e.g. the name *conTimbre*.

## ePlayer

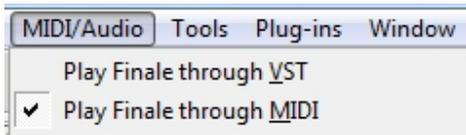
- Start ePlayer
- Define a voice in ePlayer corresponding to each voice in your score.  
The voice name should start with *IAC ePlayer*, and then an arbitrary name for your instrument, as for example:



Assign a different MIDI channel for each voice.

## Example 1: Finale™

- Start Finale™
- Select MIDI/Audio and set to Play through MIDI



- Open MIDI/Audio/Device setup/MIDI thru...  
und switch off MIDI Thru:



- Open MIDI/Audio/MIDI setup and set MIDI out to IAC-Driver conTimbre



- Open Windows/Score manager and set MIDI channels of all voices to

the appropriate values of the ePlayer orchestra.



- Select MIDI channels in the rightmost column (corresponding to the channels in the ePayer)

### Example 2: Cubase™

In Cubase™, you can play a score sending MIDI to the ePlayer via IAC-Bus. The audio output of the ePlayer can be sent to Cubase using ReWire and may be recorded.

- Start Cubase™. Select Devices/Max 6 Rewire.



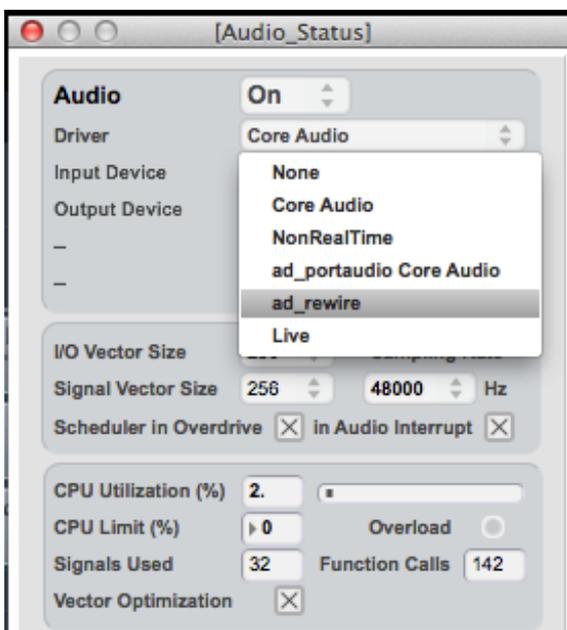
- Open Max 6 ReWire window in Cubase and switch on the desired audio output channels of the ePlayer:



In Cubase™, create the desired MIDI voices. The MIDI input has to be set to “not connected”. The MIDI output has to be set to IAC-Driver conTimbre.



In the ePlayer audio settings window, set the driver to ad\_rewire:



## Legato effect

The ePlayer plays sustained notes with a legato effect if a previous note in same voice is still playing.

## Patching audio outputs

If you wish to send audio outputs of the ePlayer to some other audio software (e.g. a hard disk recording system), we recommend to install Jack (<http://jackaudio.org>).

If the ePlayer is used within a MAX/msp environment ([www.cycling74.com](http://www.cycling74.com)), audio outputs of the ePlayer may be sent to your own patch. Use the following MAX object:

```
receive~ 1contimbre_oplayer_signal~
```

Replace the the number 1 by the number of your desired output channel of the ePlayer.

## MAX and OSC messages

If the ePlayer is used within a MAX/msp environment ([www.cycling74.com](http://www.cycling74.com)), it may receive MAX messages. All messages may be received in the same way by OSC. If using MAX messages, you have to send the messages to contimbre\_oplayer:

```
s contimbre_oplayer
```

Developer of MAX patches should avoid sends and receives with addresses which start with "contimbre". The send/receive address may already be taken by the ePlayer.

• **noteon** *voicename* *pitch* *dynamic* [*option*]

Explanation:

A note is switched on.

*voicename* is the name of the voice.

*pitch* (0.-127.) is the MIDI pitch as float. Floating point values may result in microtones.

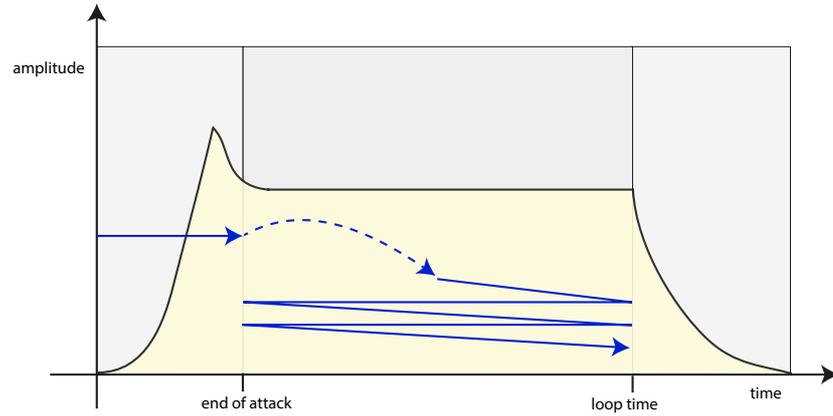
*dynamic* (0-127) is the dynamic as integer value.

*option* additional parameters, which do not have to be set necessarily.

*option* (jump\_after\_attack)

After the attack, a crossfade into a random position within the sample is executed. This may be used to avoid a quasi-mono effect that can occur when different voices are playing identical samples at the same time. For percussive sounds, the time jump ahead is at most one second.

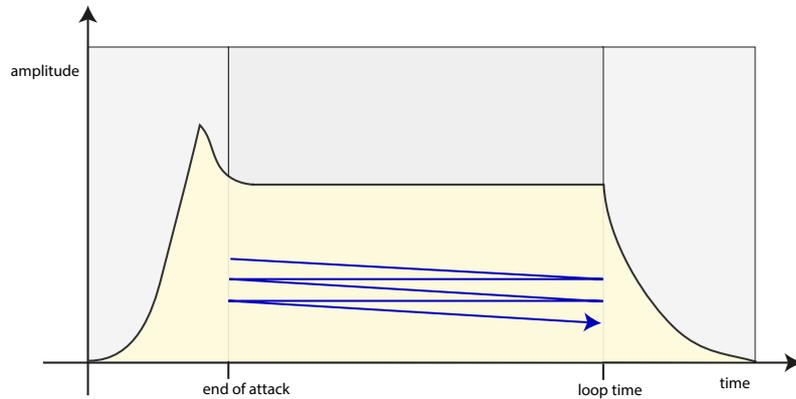
Example: noteon myvoice 60. 20. jump\_after\_attack,



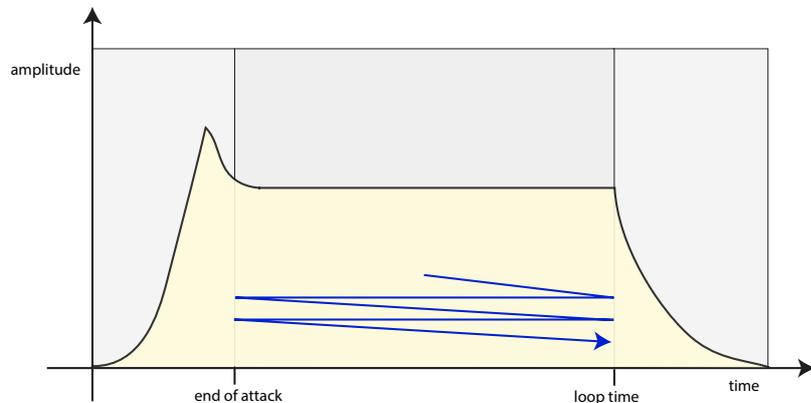
*option (time xxx)* plays the sample not from the beginning, but from a given time xxx in seconds. Percussive sounds do not react to this option.

Example: `noteon myvoice 60. 20. time 1.432`

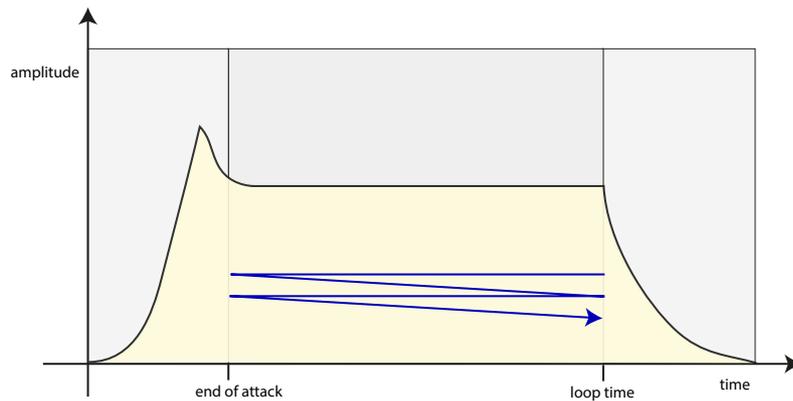
*option (time\_fraction xxx)* plays the sample not from beginning, but from a percentage of the length of the sample. A fraction of zero means play from directly after the attack. A fraction of one means play from the end of the sample. This is the same as an immediate loop jump. Percussive sounds do not react to this option.



Example: `noteon myvoice 60. 20. time_fraction 0.0`



Example: `noteon myvoice 60. 20. time_fraction 0.5`



Example: `noteon myvoice 60. 20. time_fraction 1.0`

• **`noteoff voicename pitch`**

Explanation:

A note is switched off.

*voicename* is the name of the voice.

*pitch* (0.-127.) is the MIDI pitch as floating point value of the note previously switched on.

• **`detuned_noteon voicename pitch detune dynamic [option]`**

Explanation:

Switches a note on, which is (artificially) detuned.

*voicename* is the name of the voice.

*pitch* (0.-127.) is the MIDI pitch as float, which is corresponding to the keyboard layout. Floating point values may result in microtones.

*detune* (max +24) is the detune value in semitones as floating point value.

*dynamic* (1-127) is the dynamic as integer value.

*option* additional parameters, which do not have to be set necessarily.

*option* (`jump_after_attack`) description see **`noteon`**

Example:

`detuned_noteon myvoice 60. -1.2 20. jump_after_attack,`

*option* (`time xxx`) see **`noteon`**

Example: `detuned_noteon myvoice 60. -1.2 20. time 1.432`

*option* (`time_fraction xxx`) description see **`noteon`**

Example:

`detuned_noteon myvoice 60. -1.2 20. time_fraction 0.432`

• **detuned\_noteoff** *voicename* *pitch* *detune*

Explanation:

A detuned note is switched off.

*voicename* is the name of the voice.

*pitch* (0.-127.) is the MIDI pitch as float, which is corresponding to the keyboard layout.

*detune* is the detune value in semitones as floating point value.

• **ctnote** *voicenum* *noteid* *pitch* *dynamic* *duration* [*option*]

Explanation:

Switches a note on. A note id (integer) is stored together with the note so that later ctnoteoff messages can be directed specifically to this note. The note id should be a unique number. This solves a problem which can arise when several notes with same pitch are switched on. In this case a noteoff with this pitch would just select any playing note with this pitch and switch it off.

A possible note duration is included. The quality of rapid sequences is improved using this information. If *voicenum* does not exist (see below), a new voice is created.

*voicenum* is an integer. The note is addressed to a voice with the name "voice" plus the voice number.

E.g. sending a ctnote with voice number 3 is directed to the voice with the name "voice3". If voice 3 wasn't found, ePlayer will create a new voice with this name, copying an already existing voice.

*noteid* Integer which should be a unique value. It is used to identify the note with ctnoteoff.

*pitch* (0.-127.) is the MIDI pitch as float, which is corresponding to the keyboard layout. Floating point values may result in microtones.

*dynamic* (1-127) is the dynamic as integer value.

*duration* duration of note. Unused.

*option* additional parameters, which do not have to be set necessarily.

*option* (jump\_after\_attack) description see **noteon**

Example:

ctnote 2 0 60. 20. 0. jump\_after\_attack

*option* (time xxx) see **noteon**

Example: ctnote 1 0 60. 20. 0. time 1.432

*option* (time\_fraction xxx) description see **noteon**

Example:

ctnote 1 0 60. 20. 0. time\_fraction 0.432

• **ctnoteoff**     *noteid*

Explanation:

Switches off a ctnote. The note is just selected by the *noteid*.

*noteid*                      Integer. The ePlayer looks for a playing note with this note id and switches it off.

Glissando commands

• **glissando**     *voicename*     *pitch*     *interval*     *duration*

Explanation:

executes a glissando for a note.

*voicename*                      is the name of the voice.

*pitch*     (0.-127.) is the MIDI pitch as float of the note, for which a glissando shall be executed.

*interval* (max +24) is the interval in semitones of the glissando.  
The reference is always the initial (microtonal) pitch of the note.  
If the note is already detuned upwards, the maximal positive interval may be smaller.

*duration*                      is the duration in milliseconds of the glissando.

• **ctglissando**     *noteid*     *interval*     *duration*

Explanation:

executes a glissando for a note.

*noteid*                      Integer. The ePlayer applies the glissando to a playing note with this note id.

*interval* (max +24) is the interval in semitones of the glissando.  
The reference is always the initial (microtonal) pitch of the note.  
If the note is already detuned upwards, the maximal positive interval may be smaller.

*duration*                      is the duration in milliseconds of the glissando.

• ***detuned\_glissando*   *voicename*   *pitch*   *detune*   *interval*   *duration***

Explanation:

executes a glissando for a detuned note. The note for which a glissando should be executed is identified by its pitch and its detune. The glissando value adds up to the detune value.

*voicename*            is the name of the voice.  
*pitch*    (0.-127.)    is the MIDI pitch as float of the note, for which a glissando shall be executed.  
*detune*                the detune value of the note in semitones.

*interval* (max +24) is the interval in semitones of the glissando.  
The reference is always the initial (microtonal) pitch of the note.  
If the note is already detuned upwards, the maximal positive interval may be smaller.  
*duration*             is the duration in milliseconds of the glissando.

• ***voiceglissando*   *voicename*   *interval*   *duration***

Explanation:

executes a glissando for all notes of the voice.

*voicename*            is the name of the voice.  
*interval* (max +24) is the interval in semitones of the glissando.  
The reference is always the initial (microtonal) pitch of the note.  
If the note is already detuned upwards, the maximal positive interval may be smaller.  
*duration*             is the duration in milliseconds of the glissando.

## Dynamic commands

Dynamic changes always try to simulate true crescendi or diminuendi. The timbre of the sound will therefore change within the dynamic change. The algorithm for the interpolation depends on the kind of sound and the polyphony.

• ***dynamic*    *voicename*    *pitch*    *dynamic\_target*    *duration***

Explanation:

executes a dynamic change.

*voicename*            is the name of the voice.  
*pitch*    (0.-127.)    is the MIDI pitch as float of the note, for which a crescendo or diminuendo shall be executed.  
*dynamic\_target* (0.-127.) is the dynamic target. 0. means ppp, 127. means fff.  
*duration*             is the duration in milliseconds of the dynamic change.

• ***detuned\_dynamic*   *voicename*   *pitch*   *detune*   *dynamic\_target*   *duration***

Explanation:

executes a dynamic change for a detuned note. The note for which a dynamic change should be executed is identified by its pitch and its detune.

*voicename* is the name of the voice.  
*pitch* (0.-127.) is the MIDI pitch as float of the note, for which a crescendo or diminuendo shall be executed.  
*detune* the detune value of the note in semitones.  
*dynamic\_target* (0.-127.) is the dynamic target. 0. means ppp, 127. means fff.  
*duration* is the duration in milliseconds of the dynamic change.

• **ctdynamic** *noteid* *dynamic\_target* *duration*

Explanation:

executes a dynamic change.

*noteid* Integer. The ePlayer applies the dynamic change to a playing note with this note id.  
*dynamic\_target* (0.-127.) is the dynamic target. 0. means ppp, 127. means fff.  
*duration* is the duration in milliseconds of the dynamic change.

• **voicedynamic** *voicename* *dynamic\_target* *duration*  
(deprecated. Use **volume** instead.)

Explanation:

executes a dynamic change for all notes of a voice.

*voicename* is the name of the voice.  
*dynamic\_target* (0.-127.) is the dynamic target. 0. means ppp, 127. means fff.  
*duration* is the duration in milliseconds of the dynamic change.

• **ctvoicedynamic** *voicenum* *dynamic\_target* *duration*  
(deprecated. Use **ctvolume** instead.)

Explanation:

executes a dynamic change for all notes of a voice.

*voicenum* is an integer. The dynamic change is addressed to a voice with the name "voice" plus the voice number.  
E.g. sending a ctvoicedynamic with voice number 3 is directed to the voice with the name "voice3".  
*dynamic\_target* (0.-127.) is the dynamic target. 0. means ppp, 127. means fff.  
*duration* is the duration in milliseconds of the dynamic change.

• **volume** *voicename* *volume\_target*

Explanation:

executes a volume change for all notes of a voice immediately. The dynamics of all currently playing notes of this voice stay in the same relation to each other after the volume change, but do not get softer than ppp or louder than fff.

*voicename* is the name of the voice.  
*dynamic\_target* (0.-127.) is the volume target. 0. means ppp, 127. means fff.  
The initial volume of a voice is 85.

• **ctvolume** *voicenummer* *volume\_target*

Explanation:

executes a volume change for all notes of a voice immediately. The dynamics of all currently playing notes of this voice stay in the same relation to each other after the volume change, but do not get softer than ppp or louder than fff.

*voicenummer* is an integer. The dynamic change is addressed to a voice with the name "voice" plus the voice number.  
E.g. sending a ctvoicedynamic with voice number 3 is directed to the voice with the name "voice3".  
*dynamic\_target* (0.-127.) is the volume target. 0. means ppp, 127. means fff.  
The initial volume of a voice is 85.

• **dynamic\_correction** *onoff*

Explanation:

When *dynamic\_correction* is switched on (default), all dynamics depend on the current MIDI volume value.

*onoff* 1=on, 0=off.

We recommend the default setting „on“. If you directly send MAX or OSC messages, it may be better to switch it off as dynamics will not be corrected artificially.

## Reverb commands

• **reverb\_state** *onoff*

Explanation:

Switches the reverb on/off.

*onoff* on/off. 1=on, 0=off.

• **reverb\_gain** *gain*

Explanation:

Sets the gain of the reverb

*gain* Gain in dB, 0=normal gain.

• **reverb\_out** *output onoff*

Explanation:

Sets an audio output of the reverb

*output* output channel. 0 means first channel  
*onoff* channel on/off (1/0)

• **reverb\_voicegain** *index gain*

Explanation:

Sets the gain of a voice into the reverb

*index* Index of the voice (0 means first voice in the orchestra list)  
*gain* Gain in dB, 0=normal gain.

• **reverb\_time** *time*

Explanation:

Sets the reverb time

*time* reverb time in seconds.

• **reverb\_roomsize** *size*

Explanation:

Sets the room size.

*size* room size (80=normal)

• **reverb\_damping** *damping*

Explanation:

Sets the low pass damping factor.

*damping* Low pass damping (0-0.999). 0.999 means strong low pass damping.

## Decay command

• **let\_vibrate** *midipitch fullshort*

Explanation:

Switches the decay option full/short.

*midipitch* MIDI pitch  
*fullshort* full decay (1), short decay (0)

## Miscellaneous

- **program** *voicename*    *programname*

Explanation:

Executes a program change.

*voicename*            is the name of the voice.  
*programname*        is the name of the program.

- **gain**    *voicename*    *gainvalue*

Explanation:

Sets the gain of the voice.

*voicename*    is the name of the voice.  
*gainvalue*    is the gain in dB.

- **pan** *voicename*    *panvalue*

Explanation:

Sets the pan of a voice.

*voicename*    is the name of the voice.  
*panvalue*     is the pan value(0.-1.).  
                  0.0: left  
                  1.0: right

- **solo**    *voicename*    *on/off*

Explanation:

Sets a voice to solo.

*voicename*    is the name of the voice.  
*on/off*         solo on(1) or off (0).

- **mute**    *voicename*    *on/off*

Explanation:

Sets a voice to mute.

*voicename*    is the name of the voice.  
*on/off*         mute on(1) or off (0).

• **chordon** *voicename* *list of pitches* [*option*]

Explanation:

Switches a chord on. The current program has to be in chord mode.  
A chord may be switched off by a *noteoff* command with a pitch contained in the chord.

*voicename* is the name of the voice.  
*list of pitches* is a list of MIDI pitches as integers.

*option* additional parameters, which do not have to be set necessarily.

*option* (jump\_after\_attack) description see **noteon**

Example:

chordon myvoice 60 62 65 jump\_after\_attack

*option* (time xxx) description see **noteon**

Example: chordon myvoice 60 62 65 time 1.432

*option* (time\_fraction xxx) description see **noteon**

Example:

chordon myvoice 60 62 65 time\_fraction 0.432

• **all\_notes\_off**

Explanation:

Switches all notes off.

• **kammerton** *frequency*

Explanation:

sets the standard pitch.

*frequency* standard pitch in Hz.

## Troubleshooting

Sequencer programs like Logic™, Finale™ or Sibelius™, can cause MIDI loops when playing a MIDI sequence through the IAC bus (Mac) while at the same time listening to the IAC bus (Mac). This causes ePlayer to crash. Switch off the MIDI listening function.

Program changes can cause audio clicks, if they affect a voice which is currently displayed in the program editor window or the keyboard window. To avoid this, you can for example create a new voice which is not used. Set the program window and the keyboard window to show this unused voice.

# Glossary

## **Orchestra**

An ePlayer orchestra consists of a set of voices. If a score editor is used, the score may be played with one orchestra.

## **Voice**

If a score editor is used, an eplayer voice corresponds to a voice in a score editor. A voice can receive messages (notes etc) from one MIDI device or from OSC. A voice consists of a set of programs. A voice is **not** identical to an instrument, as it may play different instruments, even within the same program. This is similar to a real musician playing on different instruments throughout a piece (e.g. a flutist can play on piccolo or on alto flute, or a pianist can play sometimes on a set of crotales).

## **Program**

A program is normally a playing mode. It contains a sound layout for a virtual keyboard or an chord mode algorithm to interpret chords as single sounds.

## **Keyboard**

A keyboard in the ePlayer always contains 128 chromatic pitches. If a MIDI note is played or a key is pressed on the keyboard, a sample is played which is bound to the specific key and the specific dynamic. When pressing a key fff a different sound may be played as when pressing the key ppp (velocity zones). If crescendi and decrescendi are interpreted, also interpolated samples may be played. The keyboard layout may contain completely heterogeneous samples, i.e. different samples from different playing modes and even instruments may be placed on different keys.

## **Instrument**

An ePlayer instrument corresponds to a real musical instrument. It can often play several hundred different playing modes.

## **MIDI Pitch**

The MIDI pitch 60 represents middle c on a keyboard. A step of one represents a semitone. Hence, 61 codes the c sharp, 62 d and 72 the c one octave higher than the middle c. Microtones may be coded with floating point numbers. 59.5 would be a quarter tone lower than the middle c.

## **MIDI Velocity**

The MIDI velocity represents the loudness of a note. The MIDI velocity 1 represents the softest possible loudness, velocity 127 the loudest possible velocity.