

# *microsynth*



**Hπ INSTRUMENTS**

Aaron Andrew Hunt

**Introduction**

<i>User Interface Basics</i> .....	3
<i>MIDI Response</i> .....	4
<i>Bug Reporting and Feedback</i> .....	4

**1. Tracks**

<i>Track Name</i> .....	5
<i>Tuning</i> .....	5
<i>MIDI Input and Channel</i> .....	6
<i>MIDI Output and Patch</i> .....	6
<i>Polyphony</i> .....	7
<i>Panning and Volume</i> .....	7
<i>Fixed Velocity MIDI Keyboards</i> .....	7

**2. Tunings**

<i>About File Types – .scl .csv .hz .tun .mtx .gly .tonex</i> .....	8
<i>Using microsynth with ScalaVista</i> .....	10
<i>Key and Map</i> .....	10
<i>Custom Key</i> .....	11
<i>Octave</i> .....	11
<i>Scale Starting Tone</i> .....	12
<i>Tuning Entries</i> .....	13
<i>Scale Editing – .scl and .tonex only</i> .....	14

**3. Switches**

<i>What is a microsynth Tuning Switch?</i> .....	17
<i>CC Switch and Switch Behaviour</i> .....	17
<i>Held Keys Behaviour</i> .....	17

**4. Tuners**

<i>What is a microsynth Tuning Tuner?</i> .....	19
<i>CC Tuner and Tuner Behaviour</i> .....	19

**5. Recordings**

<i>Renaming and Deleting Files</i> .....	20
--	----

**6. Preferences**

<i>A440</i> .....	21
<i>Import new scale files with Key and Map</i> .....	21
<i>display tuning changes in real time ... may slow down MIDI</i> .....	21
<i>get user Default tuning</i> .....	21
<i>include default Switches</i> .....	21
<i>include default Tuners</i> .....	21
<i>startup at last scene</i> .....	22
<i>Default MIDI Input</i> .....	22
<i>MIDI Input Filter</i> .....	22
<i>listen for MTS</i> .....	23
<i>External MIDI Output</i> .....	23
<i>Skip External Output Ch. 10</i> .....	23

**Credits & Resources**

## Introduction

Most software synthesizers do not allow control over the tuning of output pitches. Those which do provide tuning control usually have only the most basic and limited features. microsynth is different, providing complete control over tuning, allowing multiple tunings of up to 2048 unique pitches, options for automated switching between different tunings, and real-time changes of scale *Keys*, *Maps* and *Octaves*. Whereas the majority of software synthesizers are mono-timbral, the full version of microsynth is also 16-part multi-timbral, providing 16 tracks which can all operate independently in terms of tuning. microsynth is compatible with any MIDI controller, including Tonal Plexus and other microtonal keyboards, and can communicate with sequencing and notation software such as Sibelius, Finale, Logic, Digital Performer, etc. Output can be recorded as .wav files for importing into any DAW.

- polyphonic multi-timbral microtonal synthesizer
- 16 tracks
- unlimited tunings, any key to any pitch
- individual tuning per track
- supports MTS real-time Sysex messages
- supports .scl .tun .hz .csv .mtx .gly and .tonex tuning files
- supports single channel tuning tables of 128 tones
- supports OMNI-channel tuning tables of 2048 tones
- allows real time transposition and remapping of scales
- unlimited user-defined MIDI CC tuning switches and tuners
- comes with GM/GS Soundset
- built in reverb
- loads .sf2 soundfonts
- unlimited MIDI connections
- one virtual MIDI input and output (Mac version only)
- records wav audio files
- loads and saves scenes
- multiple Undo

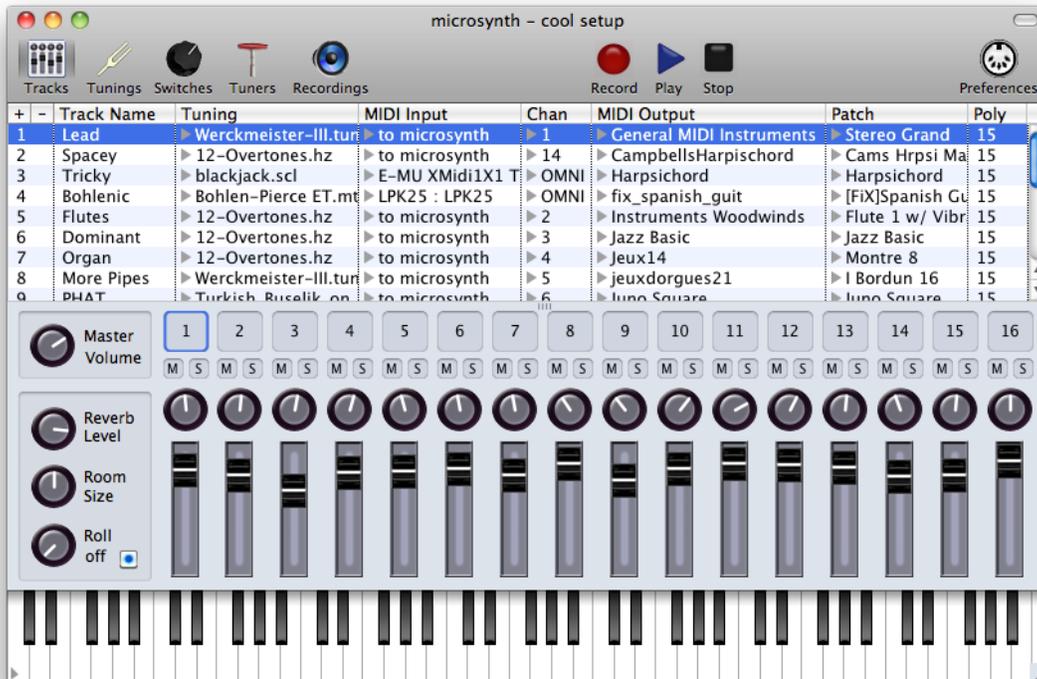
microsynth-mini is the same application in terms of tuning functions, with a few caveats:

- one track only (255 voices polyphonic, mono-timbral)
- internal synthesizer output only
- held keys behaviour when tunings change is always *none*
- no default user file option

microsynth is a stand-alone synthesizer, not a plugin.

## User Interface Basics

microsynth has a straightforward, single-window interface with a toolbar for selecting display pages and audio functions.



On each page of the interface, information is displayed primarily in lists, with selection of list items using popup menus.

A gray triangle is an indication of a popup menu.

At the bottom of the window is a keyboard having a popup triangle in its lower left corner allowing selection of its lowest displayed octave.

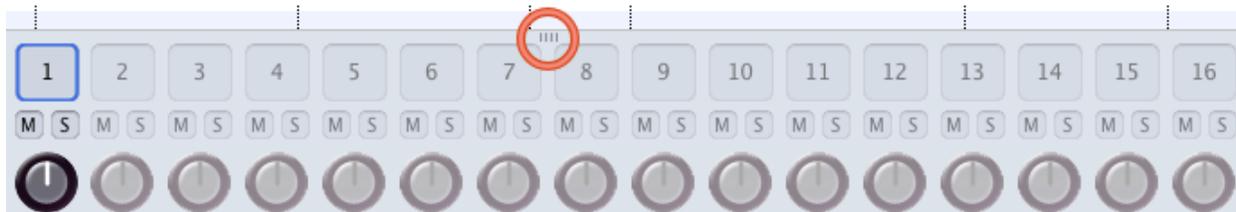




Wherever items may be added to or removed from a list, buttons marked + and - will be found in the upper left of the list header to add and remove items.

Control-Click or Right-Click a list item to *Duplicate* or *Remove* it from a list.

A small divider at the top of the mixer can be used to move the mixer out of the way. This divider moves by default on the Preferences page.



## MIDI Response

microsynth is a single-process audio application, which means that the following interface actions will affect MIDI response.

- selecting pages
- changing window position or size
- using popup menus
- moving knobs and sliders

microsynth is designed to respond to external input without the need for window interaction during audio output; changes to track volume, panning, patch changes, tuning switches, tuning key, map, and octave changes, are all designed to be automated through MIDI input, as described in this documentation. For best MIDI response, the Preferences option *display tuning changes in real time* should also be left off.

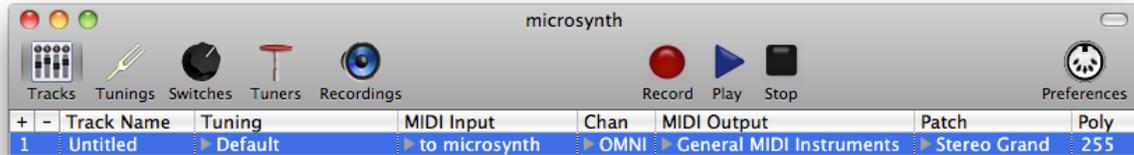
## Bug Reporting and Feedback

Please report any problems you may experience with microsynth directly by using the menu item *Report a Bug*. Before doing so, please also check the [microsynth reports webpage](#), which lists all known issues and feature requests. Please send feedback and suggestions by email directly or using the menu item *Send an Email*.

We ask kindly that you do not post complaints in Mac App Store Reviews, as we have no way of corresponding with you to correct the problem in that case. When you report bugs directly as described above, we can work with you to resolve the issue.

# 1. Tracks

As mentioned in the introduction, microsynth-mini is limited to a single track. Because of this, there is no need for a tracks list in microsynth-mini, and track information is shown in the mixer area. The multi-track version of microsynth shows track information in a list above the mixer area.



microsynth offers a basic reverb module, with adjustable level, room size, and roll off (filtering) properties.

## Track Name

Each track is initially *Untitled*, and can be changed to whatever name you like. Just double-click on the cell to edit the name. Track names do not need to be unique.

## Tuning

Each track can be assigned to any tuning, and all tunings which have been loaded into the current scene are available to all tracks. A tuning exists independent of any track, and any number of tracks may use the same tuning. For more information, see *Chapter 2: Tunings*

Each tuning assignment may also be changed in real time using *Switches*, which are discussed in *Chapter 3*.

The tunings themselves may also be altered in real time using *MTS*, discussed in *Chapter 2*, and *Tuners*, discussed in *Chapter 4*.

## ***MIDI Input and Channel***

Each track must have a unique MIDI Input. Inputs shown in the Preferences MIDI Input list are available to each track. MIDI setups can change, such that an input in a saved scene may not be currently available. If this situation arises, the track will keep the input name and simply remain silent. This way a scene will not be altered if your MIDI setup happens to change unintentionally. If you have connected a MIDI device and it does not appear in the MIDI Input popup menu, go to the Preferences page and click *Rescan MIDI*, and the input will appear, if it is in fact connected.

Each MIDI Input offers 16 channels, which can be selected individually, or all together using the OMNI option. Standard MIDI keyboards usually transmit over a single channel, while microtonal keyboards such as the Tonal Plexus transmit over multiple channels. If OMNI is selected by a track, that MIDI Input will be unavailable to other tracks, since all 16 channels will be in use.

## ***MIDI Output and Patch***

Included in microsynth is a SoundFont of 128 General MIDI Instruments. Other SoundFonts may be loaded and used as the output. There are several locations in which you may place SoundFont files so that microsynth will be able to load them:

- Library/Sounds/Banks
- User/Library/Sounds/Banks
- User/Music/microsynth/SoundFonts sf2

Output is sent to microsynth by default, which means the result of playing MIDI notes will be hearing a SoundFont retuned and rendered by microsynth. If *enable MIDI Outputs* is checked in the Preferences, MIDI output may also be sent to an external application or hardware device. When sending to a MIDI Output, the track is limited to 16 voice polyphony, and pitches are not guaranteed to be correct, because the receiving synthesizer may not be able to respond to the MIDI messages which are sent. The requirements which must be met are listed on the compatibility page of the h-pi website at <https://hpi.zentralzone.com/compatibility>

Each SoundFont contains at least one set of instrument samples which together are called a *Program* or a *Patch*. Some SoundFonts, such as the General MIDI Instruments, have many patches to choose from. The default sound set gives you 128 choices, arranged according to the General MIDI standard. If you choose to use an external MIDI output, the sounds then depend on the destination, and the patch option in microsynth will do nothing, so it is disabled for external output.

## Polyphony

The total number of voices available to microsynth at any moment is 255. These available voices are distributed as evenly as possible over the number of tracks open in a scene, so that the more tracks you have enabled, the smaller is the number of voices available to each track.

### Panning and Volume



Each track has a mixer knob and slider to control MIDI Panning and MIDI Volume respectively. These controls will automatically move to position when the corresponding MIDI messages are received on the assigned MIDI Input and Channel for each track, unless MIDI Panning and Volume have been filtered using the *MIDI Input Filter* in the *Preferences*; see *Chapter 6: Preferences* for details.

### Fixed Velocity MIDI Keyboards

Some MIDI Keyboards, such as H-Pi Instruments Tonal Plexus TPX and U-Plex keyboards, do not have velocity sensing keys. In such cases, it is helpful to have manual control over the velocity of the MIDI notes output from the keyboard. The Volume slider can be repurposed for *Fixed Velocity* by Control-clicking or Right-Clicking the slider and choosing *Fixed Velocity* from the popup menu. After *Fixed Velocity* has been selected, moving the slider up or down changes the MIDI velocity value of incoming note messages from the MIDI keyboard.

To make fixed velocity input sound more natural, the velocity values may also be slightly randomised. To do this, after selecting *Fixed Velocity*, another Control-click will show a further option to *Humanize Velocity*. After this is selected, the incoming note messages will randomise within 10 values of the fader position.

## 2. Tunings

In microsynth, the word “tuning” means a set of tones used to make music. In the MIDI world, tunings come in groups of 128 tones, because each MIDI channel is limited to 128 MIDI Notes. microsynth handles tunings with up to 2048 tones in them, by using multiple MIDI input channels. To create tunings with more than 128 tones in them, you can use [Universal Tuning Editor \(UTE\)](#) software.

In microsynth, a scale is considered a smaller set of tones from which a larger set of tones called a tuning is derived. A tuning does not necessarily include a scale; it can just be a collection of tones with no discernible subsets. This distinction between scale and tuning is useful when getting to know different tuning file types and how they work in microsynth.

A tuning and a scale are not considered to be the same thing in microsynth.

### **About File Types – .scl .csv .hz .tun .mtx .gly .tonex**

microsynth supports seven types of tuning files. The file extension must be included with the file name in order for the file to be visible to microsynth.

.scl	text list of tones as scales in ratio or decimal form
.csv	text list of 128, 512, or 2048 MIDI Note + 14-bit Pitch Bend values
.hz	text list of 128, 512 or 2048 Hz values
.tun	text list of 128 cents values
.mtx	text list of frequency values with other special information
.gly	binary file exported from <i>Absynth</i> (Native Instruments)
.tonex	text list of scale tones in a variety of formats using xml tags

### **.scl**

The Scala format is by far the most widely used and widely available tuning files. For details on this format and how to use it, see [http://www.huygens-fokker.org/scala/scl\\_format.html](http://www.huygens-fokker.org/scala/scl_format.html)

### **.csv**

This is a file format used by H-Pi Instruments [Custom Scale Editor \(CSE\)](#), [Tonal Plexus Editor \(TPXE\)](#), and [Universal Tuning Editor \(CSE\)](#) software. These files consist of comma-separated lists of MIDI bytes as MIDI NOTE, PITCH BEND MSB, PITCH BEND LSB. In a .csv file, there is no information concerning a scale as such. The files may list 128, 512, or 2048 values. A list of 128 values is a tuning for one MIDI channel. A file listing 512 values covers 4 MIDI channels of 128 notes each. The 512 tone files are used by [CSE](#) for Tuning Box TBX1 hardware, which supports tunings in four layers, one layer per MIDI channel. 2048 values covers all 16 MIDI channels, used by [TPXE](#) for Tonal Plexus keyboards, where each physical octave of the keyboard is mapped to two MIDI channels.

### **.hz**

This file format was introduced by H-Pi Instruments **H-Pi Lo-Fi Microstudio (HPLF)**, also supported by [CSE](#), [TPXE](#), and [UTE](#) as an export format. They are text files consisting of a simple list of frequency values in Hz. The numbers of tones listed in the file correspond to the same options listed above for .csv files; that is, 128, 512, or 2048. As in .csv files, a .hz file contains no information about a scale as such.

### **.tun**

These files are also known as VAZ-Anamark files, and can exist in a few different varieties, but the majority of these file consist of a list 128 cents values. The cents values are given in reference to the lowest MIDI note 0, and this form of the file contains no scale information as such. More information on this format can be found at [http://www.mark-henning.de/download\\_e.php#Tuning](http://www.mark-henning.de/download_e.php#Tuning).

### **.mtx**

These files are created by Max Magic Microtuner software. Scale information can be extracted from these files. Details on this format can be found at <http://digilander.libero.it/microtuner/MicrotunerFormat.pdf>.

### **.gly**

These files are created by *Native Instruments* Absynth software, and the format is also exported by [Scala](#) (by Manuel Op de Coul), [UTE](#) and [CSE](#). Because .gly is a binary file, there are no public details on the file type, and it is impossible to make the file as text or to view the file contents as text, except in a hex editor. The file data itself is an encrypted list of 128 Hz values, one value per MIDI note, with no explicit scale data included.

### **.tonex**

This is an experimental filetype supported by microsynth, which uses xml tags to specify different aspects of a tuning. microsynth supports simple .tonex files which list tones in scale order using *tuning entries*, as discussed in this chapter below. Scale information is included in these simple .tonex files, so *Tonic*, *Key* and *Map* can all be manipulated easily.

Two sample files of each supported file format are included with microsynth. The files are installed in *User/Music/microsynth* in the following folders:

- Tunings scl
- Tunings csv
- Tunings tun
- Tunings hz
- Tunings gly
- Tunings mtx
- Tunings tonex

### **Using microsynth with ScalaVista**

*ScalaVista* is an application by H-Pi Instruments which allows you to easily browse the Scala archive of over 4000 scales. With *ScalaVista*, you can find a scale in the database and send it directly to microsynth, saving you the trouble of having to import the .scl file in microsynth.

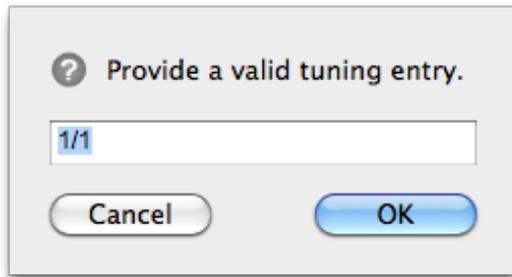
### **Key and Map**

These options allow you to change the way a scale appears on the keys of a MIDI keyboard. *Key* refers to the tonic of the scale, and *Map* refers to the physical key of a MIDI keyboard. You can also think of the *Map* as referring to an abstract MIDI Note number.

When *Key* and *Map* are the same, the given tonic at a note name should sound on the given MIDI key. For example, by default both *Key* and *Map* are set to C. This means that the tonic of the scale is C, and it should appear on the MIDI keyboard's white C key. If you instead want the key A of the keyboard to play the note A as the tonic of the scale, then select *Key* = A and *Map* = A. If you wanted the tonic to be A, but wanted this tone placed on the keyboard key C, then you would choose *Key* = A and *Map* = C.

Scales having more or fewer than 12 tones will not place the tonic note on every same key of the keyboard per octave. If you want to create your own tunings, you should use an editor like [Universal Tuning Editor \(UTE\)](#) or [CSE](#) and send the tuning to microsynth.

## Custom Key



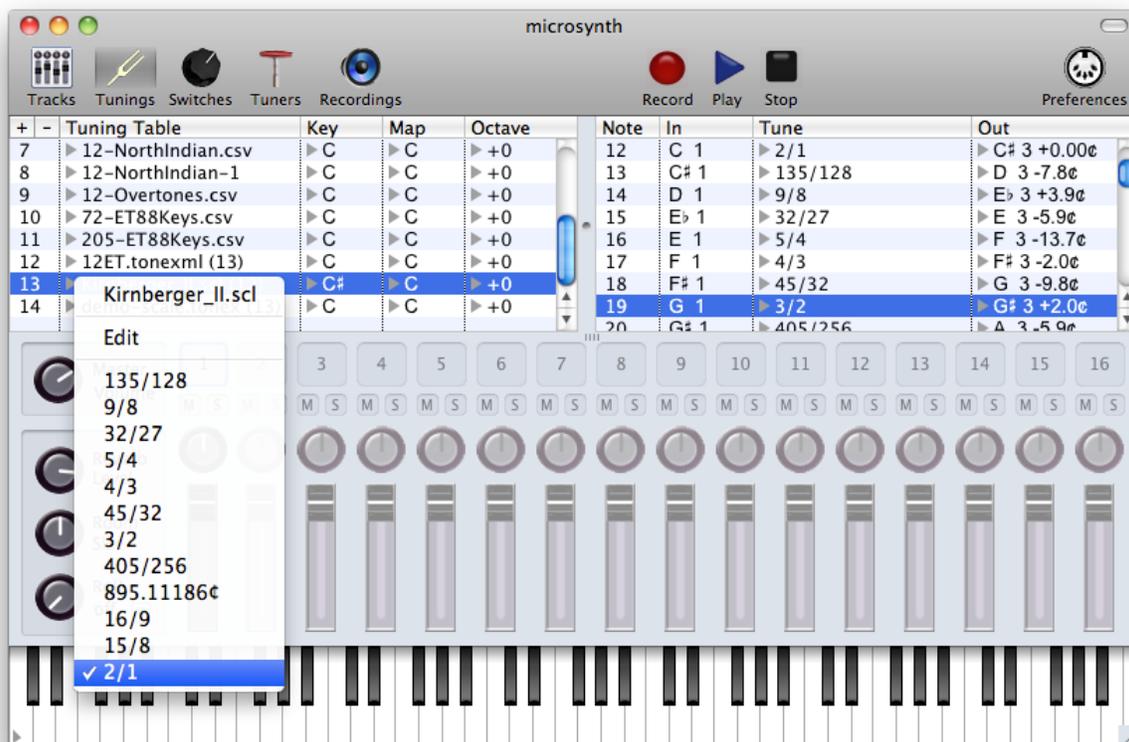
The normal *Key* options correspond to the twelve tones of 12ET, because these are the tones expected on a standard MIDI keyboard. If you want to transpose the scale instead by an interval that is not one of those twelve tones, the *Custom Key* lets you do this.

When you select this option from the popup menu, an input window will appear in which you are expected to type a valid tuning entry, which is explained in this chapter below.

## Octave

A tuning maybe transposed up and down by octaves (1:2 or 2/1 in *Just Intonation*). MIDI allows for roughly 10 octaves of tones. Different scales will map differently to the number of available MIDI keys (128), and some scales will reach the maximum MIDI note (127) fairly quickly, while others will not reach it unless you transpose the tuning up by octaves. Scales do not need to repeat at the octave; they may repeat at any interval. The interval at which a scale repeats is called its *period*. Scales do not have to have twelve tones in them; they can have any number of tones. So, the period of the scale and the number of tones in the scale both determine where the scale will be initially mapped in terms of its octave register. Bearing this in mind, to get some tunings to sound at the register you expect for music making, you will often need to transpose the tuning up or down by octaves.

## Scale Starting Tone



Scales usually start on an initial tone called the *tonic* (or 1/1, or 1:1 in *Just Intonation*). This tone does not necessarily need to be treated as the first tone in the scale. microsynth lets you select any tone of the scale as its starting tone, a practice which allows scales to be treated as *modes*, where the first tone is not necessarily the most central tone in a given musical context, as in early Western music.

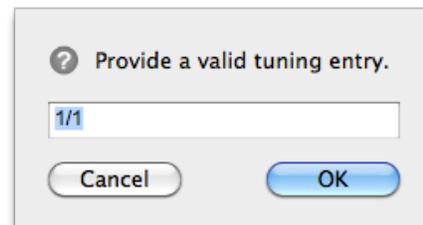
Only three of the seven supported tuning file types include scale information: .scl, .mtx, and .tonex files. Scales loaded from these files are visible in a popup menu from the Tunings list. The tone with the check mark is treated as the tonic of the scale. Tonic is always the last tone in the list by default, since the last tone specifies the *period* of the scale. The period is usually 2/1 or 1200¢ (an octave), but not always. Sometimes the period is an interval very close to 2/1, either slightly smaller or slightly larger, so that successive transpositions of the scale gradually are sharper or flatter. The period may also be a completely different interval, such as in the Bohlen-Pierce scale, which has a period of 3/1 instead of 2/1.

File formats which do not include scale information have no tonic tone; however, you can still change the *Key* and *Map* options for such scales. The entire tuning is then treated as one large scale, which is transposed and moved as you specify. Keep in mind that for these kinds of non-

scale tunings, when you move them around with the *Map* parameter, you can actually lose tones at the top and bottom of the tuning. microsynth keeps a copy of the original tuning when you perform a remapping operation, so that tones should never actually be lost.

## Tuning Entries

5	F 0	▶+533.410	▶ F 1 +31.1¢
6	F# 0	▶+598.860	▶ F# 1 -3.4¢
7	G 0	▶+705.541	▶ G 1 +3.3¢
8	G# 0	▶+812.223	▶ G# 1 +9.9¢
9	A 0	▶+918.905	▶ A 1 +16.6¢
10	B <sub>b</sub> 0	▶+1025.587	▶ B <sub>b</sub> 1 +23.3¢
11	B 0	▶+1132.269	▶ B 1 +30.0¢
12	C 1	▶+1197.719	▶ C 2 -4.6¢
13	C# 1	▶(+106.682)+1197.72	▶ C# 2 +2.1¢
14	D 1	▶(+213.364)+1197.72	▶ D 2 +8.8¢



A tuning entry in microsynth is an expression which represents a tone in different ways using math. Examples of valid tuning entries are shown below. Parentheses must be used when there are 2 terms and some operator, as shown in the last three examples.

- **1.2365256341287** ... decimal values
- **2:3** ... interval ratios a:b, where a < b
- **3/2** ... tone ratios a/b, where a > b
- **2,12** ... a degree of an equal division of an octave, where the first degree is 1
- **3^7** ... exponents, such as 3 to the 7th power
- **+35.2** ... an interval up, in cents
- **-4.3** ... an interval down, in cents
- **f=261.6255653** ... Hz values, within an available range of 7.9430 to 12911.4169 Hz
- **(1.232)\*(2:3)** ... is 1.232 transposed up by 2:3
- **(4/3)/(1.112)** ... is 4/3 transposed down by 1.112
- **(5/4)+3.1** ... is 5/4 transposed up by 3.1 cents

To force a pitch to a given octave, at the end of the entry, type Option-O (“oh”, not zero) which is  $\emptyset$ , followed by an octave number (0-10). Example: (3,12) $\emptyset$ 4 is D4 in the Key of C.

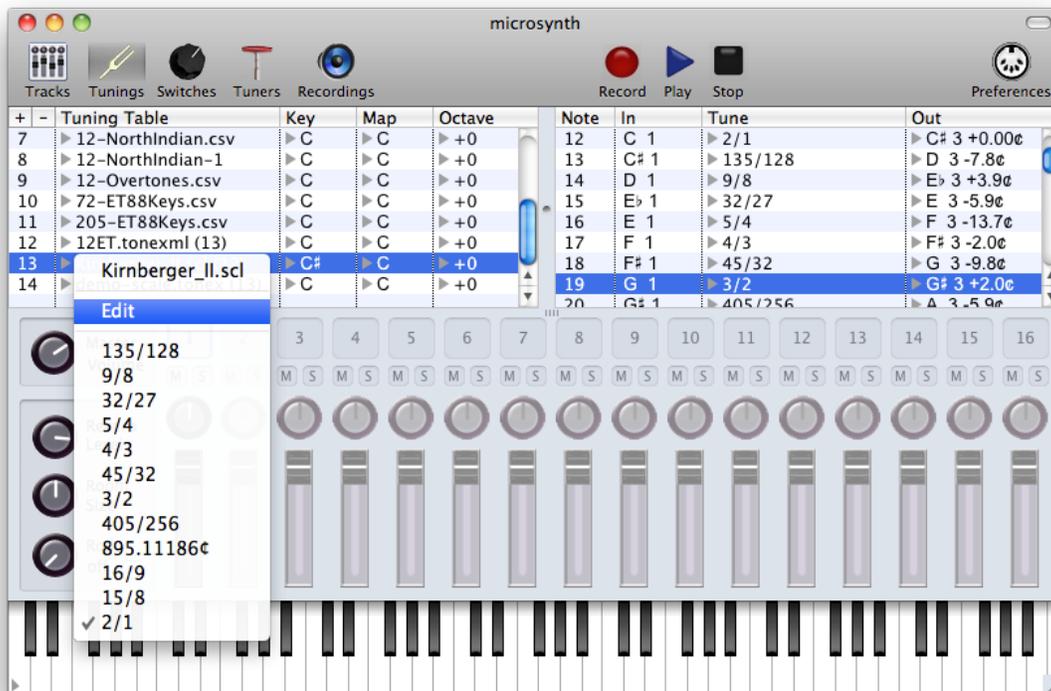
Up to 32 operations can be strung together in an entry using parentheses. Cents transposition must always be at the end in an entry using parentheses. An example semi-complex tuning entry using parentheses is given below.

**(4,13)/(11/8)\*(9/8)-23.1**

This is 4th degree of 13ET transposed down by 11/8, then transposed up by 9/8, lastly transposed down by 23.1 cents.

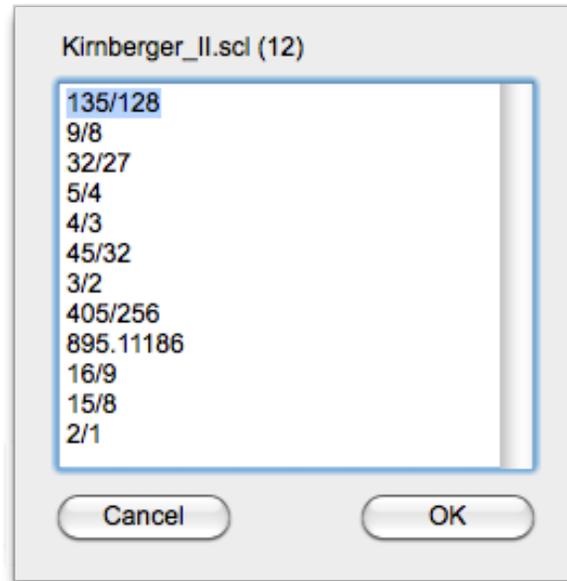
The .tonex file format is a language for tunings which supports the above syntax. See sample files installed in *User/Music/microsynth/Tunings tonex* for examples. More advanced options for tuning entries, including constants, functions, scale references and code snippets, all of which allow you to create your own scales in many different ways, are available in [Universal Tuning Editor \(CSE\)](#), [Custom Scale Editor \(CSE\)](#) and [Tonal Plexus Editor \(TPXE\)](#) software.

### Scale Editing – .scl and .tonex only

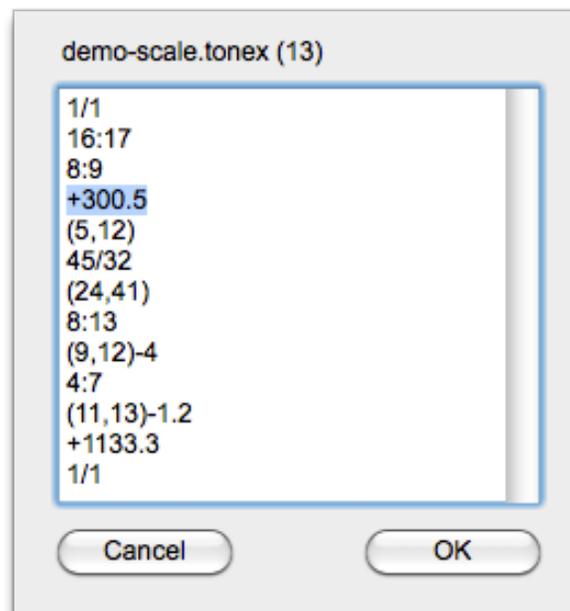


Tunings which have been imported from .scl and .tonex files include scale data which can be edited directly in microsynth. While .mtx files can include scale data, that data is formatted in a such a way that it cannot be edited in microsynth. To edit a .scl or .tonex scale, select the *Edit* option from the Tuning Table popup menu on the Tunings page.

For example, if the example scale *Kirnberger\_II.scl* has been imported, and Edit is selected, the following window appears.



When editing a .scl scale, only valid Scala entries are allowed. Note that 1/1 is not included in a .scl scale. The tonic 1/1 can be omitted or included in a .tonex file. When included, the tonic is placed both at the beginning and the end of the list of tones in the scale. When editing .tonex files, only valid tuning entries, as described in this chapter above, are allowed. The example .tonex file demo-scale.tonex is show below with several valid edits.



## Live Retuning with MTS

If *listen for MTS* is checked in the Preferences, microsynth will respond to the following MIDI Tuning Standard (MTS) System Exclusive MIDI on all input ports and channels:

- Bulk Tuning Dump (non-real-time)
- Single-Note Tuning Change (real-time)

The Bulk Tuning Dump is a list of 128 values, to retune an entire tuning table at once, whereas the Single-Note Tuning Change sends any number of arbitrary values to specified registers of a tuning table.

Note that MTS works only for tables having 128 values, so this method is not recommended for working with large tuning tables such as those used by the Tonal Plexus having 2048 values.

Once an MTS tuning change has been received, the tuning table changes from its initial format to *mts* format, and all tuning entries are expressed as deviations from 12ET.

So, if you begin with a *.scl* or *.tonex* file having its tones expressed as ratios, those expressions will be lost when you alter the tuning with MTS messages.

For details on how to create MTS messages, see <http://www.midi.org/techspecs/midituning.php>.

## 3. Switches

### *What is a microsynth Tuning Switch?*

Each track in microsynth is assigned a tuning, which you may want to change in real time. Switches in microsynth allow you to switch between any currently loaded tuning during performance or playback from a sequencer or notation program using MIDI Continuous Controller (CC) messages. You control the behaviour of the switch CC as well as the behaviour of the pitches currently sounding when the tuning switch takes place.

A tuning switch affects only the track to which it is assigned.

### *CC Switch and Switch Behaviour*

MIDI CC messages have three bytes: CC Channel, CC Number, and CC Value. The parts of a CC message relevant to a microsynth switch are the Number and Value. Select a CC Number by clicking the triangle in the *MIDI CC Switch* column. The CC Value may or may not be necessary, depending on the switch behaviour you choose. Select the switch behaviour in the second column. There are four possible switch behaviours.

- select tuning by number [data 1-127]
- select next tuning in list
- select previous tuning in list
- select tuning by name

Only when selecting a tuning by number is the CC Value used. In the three other types of switch behaviours, although the Value byte is sent according to the MIDI specification, it is ignored by microsynth.

### *Held Keys Behaviour*

Pitches sounding when a switch occurs are referred to as held keys in microsynth. Held keys may be handled in three ways.

- none
- re-bend same MIDI notes
- restrike

*None* means that sounding pitches continue sounding as they are until they are released.

*Restrike* means that all pitches sounding will immediately released and initiated at the same MIDI input note, with the same velocity as the initial Note ON message, to sound the pitch stored at that register in the newly switched tuning.

*Rebend same MIDI notes* means that pitches sounding from the previous tuning will bend into the pitches of new tuning, but only when both the input MIDI note and the output MIDI note is the same in the previous tuning and the new tuning. It is not possible to bend pitches from one tuning into another when the input and output MIDI notes are not the same in both tunings.

## 4. Tuners

### *What is a microsynth Tuning Tuner?*

In some cases you may want to change the octave, key, or map parameter of a tuning without switching to a different tuning. Tuners in microsynth allow you to alter these settings for any tuning assigned to an active track during performance or playback from a sequencer or notation program using MIDI Continuous Controller (CC) messages. You control the behaviour of the tuner CC as well as the behaviour of the pitches currently sounding when the tuner is activated. Unlike switches, which affect only a given track, a tuner will affect all tracks referencing the given tuning.

A tuner alters a tuning directly, so that all tracks referencing that tuning will immediately reflect changes made by the tuner.

### *CC Tuner and Tuner Behaviour*

There are three bytes in a MIDI CC message: CC Channel, CC Number, and CC Value. The parts of a CC message relevant to a microsynth tuner are the Number and Value. Select a CC Number by clicking the triangle in the *MIDI CC Tuner* column. The CC Value varies depending on the tuner behaviour you choose. Select the switch behaviour in the second column. There are six possible tuner behaviours.

- select Key & Map [data 0-11]
- select Key [data 0-11]
- select Map [data 64 = 0]
- select Octave [data 64 = 0]
- transpose Octave up
- transpose Octave down

The CC Value is used in two ways by MIDI standards: either as a positive value between 0 and 127, or as a centred value including negative numbers, where the middle number 64 = 0, numbers above are positive (65 = 1), and numbers below are negative (63 = -1).

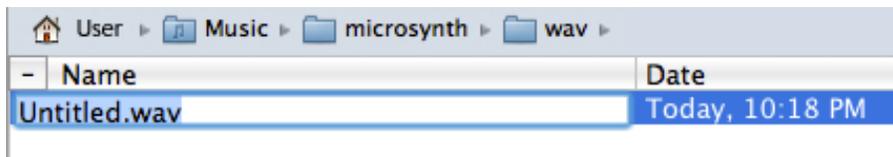
In the two tuner behaviours involving the *Key*, the CC Value is expected to be between 0 and 11, corresponding to twelve tones of a MIDI octave. Changing the *Map* or *Octave* may be a negative value, so CC Value 64 is used as 0. The Value byte is ignored by microsynth when the Octave up or Octave down behaviours are selected.

## 5. Recordings



The microsynth transport allows you to record and play back wav audio files, and the recordings page lists those files with their creation dates and durations. A file can be played back while recording a new file; however, running other processor-intensive software while recording in microsynth may result in dropouts.

### *Renaming and Deleting Files*



To rename a file, just double-click the Name column and type a new name. Typing the .wav extension is not necessary, as microsynth adds it for you.

To delete a file, select it and type the delete key, or click the minus button.

Deleting a recording is an action that cannot be undone.

## 6. Preferences

### **A440**

The default reference pitch of microsynth is A4 sounding at 440 Hz. This pitch may be adjusted up or down in cents. Note that tuning tables may or may not expect a reference tone of A440 Hz, and tunings which use Hz values in .tonex format (f=HzValue) may give incorrect frequencies if you change the setting of the reference tone to something other than A440. Note also that not all SoundFonts are recorded at exact reference pitches, and tuning accuracy may vary from one SoundFont to another.

### ***Import new scale files with Key and Map***

The values you choose here will be used when importing tuning files which include scale data: .scl, .tun, and .tonex tuning files. Setting these values does not change any currently loaded tunings. Tuning formats other than .scl, .tun, and .tonex do not apply the *Key* and *Map* parameters on import, although you may apply them after importing. To change an existing *Key* and *Map* of a loaded tuning, use the popup menus of the tunings list.

### ***display tuning changes in real time ... may slow down MIDI***

If you are sending many switch and tuner CC messages to microsynth, you may find that MIDI traffic begins to slow down a bit. Usually this will happen only if you use switches and tuners heavily. In such a situation, turning off real time display updating will improve MIDI response.

### ***get user Default tuning***

You may create a .microsynth file to serve as your default template when *File > New* is selected. Just save a microsynth file in the location *User/Music/microsynth/Defaults*, and check this option in the Preferences. If this option is not used, a new file have one track called *Untitled*, and one tuning called *Default*, with 4 default switches and 6 default Tuners, if you have those options checked.

### ***include default Switches***

When you add a new track to microsynth, that track may include a default set of switches, allowing you to control the tuning assigned to the new track. If you do not want new tracks to have a default set of switches, uncheck this option.

### ***include default Tuners***

Adding a new track to microsynth may also include a default set of tuners, allowing you to control the octave, key, and map of the tuning assigned to the new track. If you do not want new tracks to have a default set of tuners, uncheck this option.

## ***startup at last scene***

When this is selected, microsynth will store the current scene internally and open in the state it was last closed by loading the internally saved scene. If you use this option, be aware that microsynth does not automatically update any external files when quitting, and does not load an external file when starting up. To save the scene externally, you must select *File > Save Scene*, and give the file a name and location.

## ***Default MIDI Input***

If you have a Network MIDI connection, it will appear in this list. microsynth creates a virtual MIDI input which also appears here. Any other MIDI input device or virtual port which is available will appear in this list. If you have connected a MIDI device and it does not appear in the MIDI Input list, click *Rescan MIDI*, and the input will appear, if it is in fact connected.

## ***MIDI Input Filter***

microsynth is capable of responding to the following standard MIDI messages on all active tracks.

- Note ON / OFF
- Sustain ON / OFF
- Pitch Bend Change
- Patch Change
- Volume CC
- Panning CC
- Modulation CC
- Assigned Switch CCs
- Assigned Tuner CCs

When microsynth receives Pitch Bend Changes, the tunings of the sounding pitches on the given track are retained as those pitches bend, and pitches will not bend beyond a bend limitation of +/- one 12ET semitone.

The tuning deviation of a given MIDI note from the default MIDI 12ET tuning limits its Pitch Bend range to +/- one 12ET semitone minus its tuning deviation.

For example, a note is tuned as  $C4+33\text{¢}$ . This note has an upward bend range of  $100-33\text{¢} = 67\text{¢}$ , and a downward bend range of  $33-100\text{¢} = -67\text{¢}$ .

Volume and Panning changes will move the sliders and knobs on the microsynth mixer unless you uncheck those options in the MIDI Filter list.

MIDI messages in the filter list other than Patch, Volume, Panning, and Modulation are unsupported by the microsynth synthesizer.

The filter list is there primarily for controlling how microsynth responds to MIDI Volume, Panning, and Patch Change messages. Other CC messages are listed for use when the external MIDI output option is used.

As of version 1.3.6, microsynth stores filter lists with scenes.

### ***listen for MTS***

When this option is checked, microsynth will respond to incoming MIDI Tuning Standard (MTS) System Exclusive MIDI messages on every active input port in addition to standard MIDI messages. See *Chapter 2: Live Retuning with MTS* for details.

### ***External MIDI Output***

A list of MIDI outputs, including a microsynth virtual port, is shown in this list, and is available as an output option in the multi-track version of microsynth when *enable External MIDI Output* is selected. If a track is assigned to a MIDI output, it means that the internal microsynth synthesizer will not sound for that track, and some external destination is intended. The quality and tuning accuracy of an external output is not guaranteed by microsynth, because the application knows nothing about an external destination beyond its MIDI connection details.

microsynth sends external MIDI to multiple output channels, so when sending a track to external output, be sure to enable OMNI input in the receiving synth, even if the music you send over the track is monophonic. If the receiving synth is monophonic, results may be unpredictable.

### ***Skip External Output Ch. 10***

External MIDI destinations may be GM compatible, in which case it will designate its channel 10 for percussion sounds only. microsynth can skip channel 10 in this case. If you hear a percussion sound ever so often when sending MIDI to an external output, check this option and the percussion sound should go away.

## Credits & Resources

microsynth was initially commissioned by [John Reed](#), inventor of Diano music software.

All versions of microsynth are designed and programmed by Aaron Andrew Hunt, using [Xojo](#) and [MBS Plugins](#) on a [Mac](#).

Special thanks to [Felix Krauss](#), author of the original FluidSynth wrapper for REAL Studio.

The General MIDI Instruments SoundFont included with microsynth is a version of GeneralUser GS SoundFont, provided courtesy of [S. Christian Collins](#).

Many thanks to the [FluidSynth](#) team for all their hard work, and for providing the FluidSynth library under the LGPL v2 license.

This documentation was written by Aaron Andrew Hunt, using Apple [Pages](#).

Resources for Tunings and SoundFonts are available through the menu items *H-Pi Instruments > Get Tunings* and *Get SoundFonts*.

Thank you for supporting H-Pi Instruments and microsynth.

© H-Pi Instruments · FOR THE FUTURE OF MUSIC