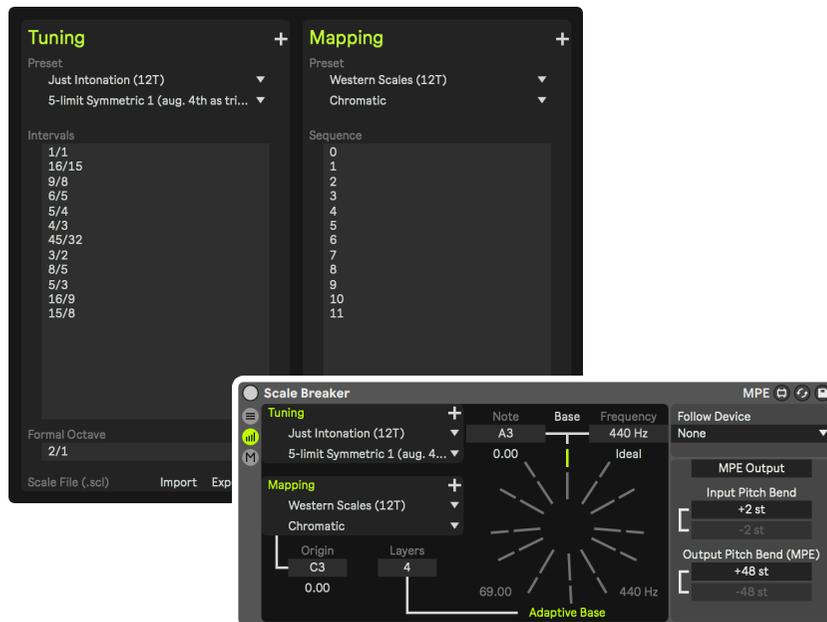




Table of Contents



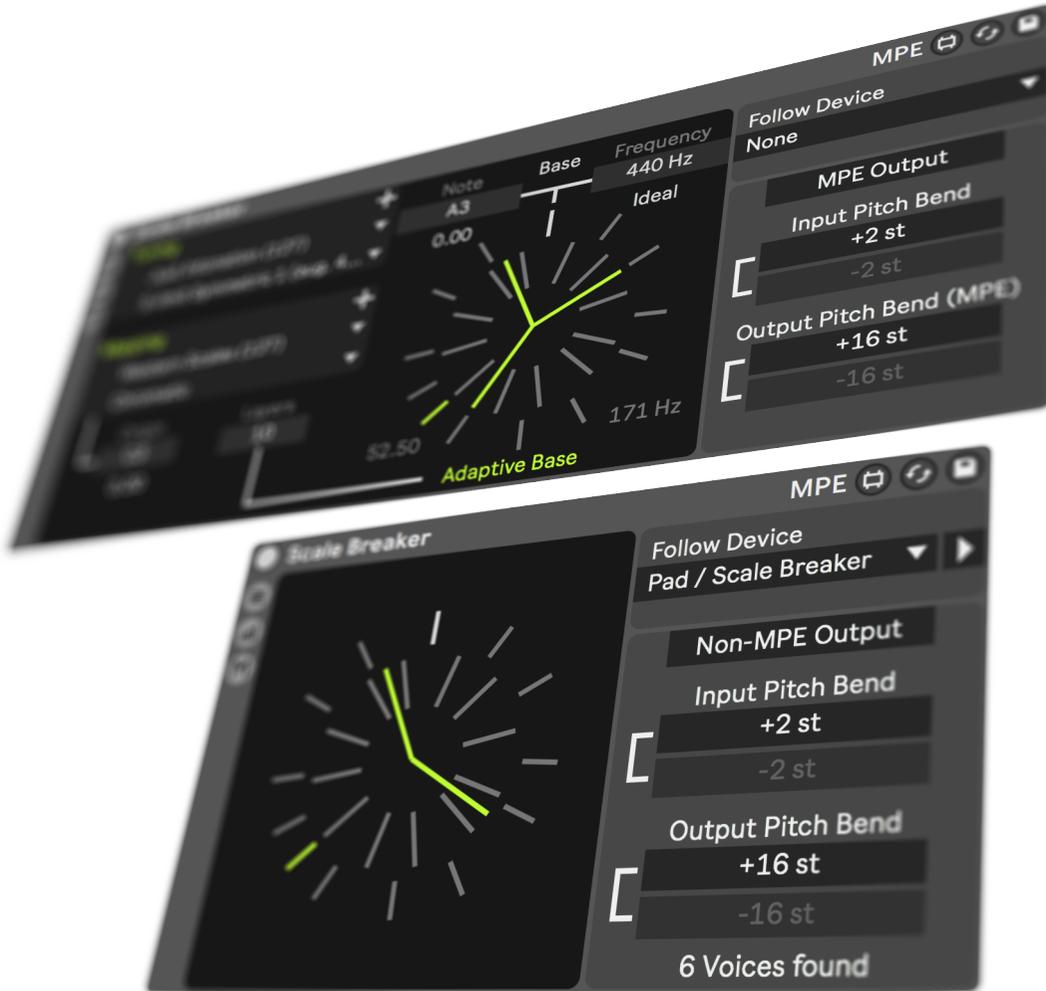
| | |
|---|----|
| Introduction | 3 |
| Setup | 4 |
| Connect to MPE Instruments | 4 |
| Connect to Non-MPE Instruments | 5 |
| Connect to Non-MPE Instruments polyphonically | 6 |
| Usage | 7 |
| Overview | 7 |
| Adaptive Base | 8 |
| Follow Device | 9 |
| Tuning & Mapping Slots | 10 |
| MPE Control | 11 |
| Freeze | 12 |
| Global Preferences & Advanced Settings | 13 |
| Tuning & Mapping Editor | 14 |
| Custom Tunings | 15 |
| Custom Mappings | 16 |



ChaosCulture.net



IsotonikStudios.com



Welcome to **Scale Breaker**, a Max for Live device for retuning MIDI controlled instruments in **Ableton Live** 10 and 11.

Live, as well as most **VST Instruments** and **DAWs**, is tied to the standard **12-Tone Equal Temperament Tuning** that is almost exclusively used in today's music. This device modifies the **Pitch Bend** data of **MIDI** signals to break out of this limitation, forcing the instruments to play in tunings that otherwise would be impossible. **Scale Breaker** is packed with a lot of small functionalities in order to make it the ultimate retuning device.

Pitch Bend data is not only being offset by a static value once a new note is played, it is also translated continuously. Pitch envelopes of **Live's MPE MIDI Editor** or the input of **MPE MIDI Controllers** are converted entirely. Automations related to the tuning are latency compensated and will update all currently held notes.

Additionally to the **Tunings**, this device supports custom **Keyboard Mappings**. It can import and export **.scl** and **.kbn** files, and a built-in editor can be used to make customizations in real time.

An **Adaptive Base** allows **Just Intonation** tunings to be used properly in any key, regardless of the played chords and their root notes. Such a system is necessary and the reason why **12-TET** became the standard, as it doesn't require retuning the instruments for almost every chord that is being played. However, retuning gets easier with **Scale Breaker**, since the **Adaptive Base** is controlled with low velocity notes, making it possible to specify **Base Tunings** within **Live's MIDI Editor** or through other means of generating **MIDI** notes - such as **MIDI Keyboards** and other plugins. There is no need to use additional controls in a separate window.

The **Follow** system can be used to link together the **Tunings** and **Mappings** of some or all instruments in a song. This includes the **Adaptive Base**, as well as automations of the **Base Tuning**. All of this data is communicated across the devices in real time with latency compensation for most accurate timings. It also helps to write music in **Just Intonation**, as every instrument is guaranteed to be in tune with each other for every played chord.

Tuning & Mapping Slots can be used to load multiple **Tunings** and **Mappings** into the device. Control which one should be used with automation or with **MPE Note Expressions** on a per note basis.

Finally, the **Freeze** functionality can be used to create smooth transitions. Switch between different **Tunings** for each individual note or for the whole **Adaptive Base** system directly from **Live's MIDI Editor**.

From basic **Temperaments** and **Microtonal** scales with any amount of notes or formal octave intervals to **Adaptive Just Intonation** and complete experimental setups. This device aims to provide everything you need to control any **MIDI**-controlled instrument, such as stock **Live** devices, **VST Instruments** or hardware instruments, whether they support **MPE** or not, and still keep them polyphonic.



- 1 Make sure the MPE Toggle is set to **MPE Output**.
- 2 The **Output Pitch Bend** range needs to be set to the same value as the MPE pitch bend range in your instrument.

48 semitones is the standard range almost all instruments use. Note that this is the **MPE** pitch bend, not the regular one that is typically controlled with the pitch bend wheel on **MIDI keyboards**.

The **Output Pitch Bend** parameters will be highlighted for a short time if the device can't produce the desired frequency due to the range being too small.

The **Input Pitch Bend** range is used to specify how **Scale Breaker** will interpret the incoming **MIDI** data for the regular pitch bend. By default, this device will convert it into **MPE** Pitch Bend individually for each note. This behavior can be deactivated in **Options**.



Scale Breaker

Tuning

- Temperaments (12T)
- Equal (12 EDO)

Mapping

- Western Scales (12T)
- Chromatic

Origin: C3, 0.00

Note: A3, Base: 440 Hz, Frequency: Ideal

Adaptive Base

MPE

Follow Device: None

Non-MPE Output **1**

Input Pitch Bend: +2 st, -2 st

Output Pitch Bend **2**: +2 st, -2 st

Monophonic Output

1 Make sure the MPE Toggle is set to **Non-MPE Output**.
Only required for Live 11 users. This toggle doesn't exist on Live 10, since MPE is not supported by it.

2 Make sure the **Output Pitch Bend** is set to the same value as the pitch bend range in your instrument.

Contrary to connecting to **MPE** instruments, this is the regular pitch bend that is typically controlled by the pitch bend wheel on **MIDI** keyboards.

The **Output Pitch Bend** parameters will be highlighted for a short time if the device can't produce the desired frequency due to the range being too small.

The **Input Pitch Bend** range is used to specify how **Scale Breaker** will interpret the incoming **MIDI** data. Some tunings require the **Output Pitch Bend** to have a larger range in comparison.

Operator

Coarse, Fine, Fixed, Level

Velocity, Key, Aftertouch, Pitch Bend, Mod Wheel

Conn. A, Amt. A, Conn. B, Amt. B

Pitch Env: 0.0%, 0%, 0 st

Pitch Bend: +2 st

Voices: 6, Rtg, Interpol., Antialias, Pan, Pan<Key, Pan<Rnd

LFO: Sine, Rate: 64.00, Amount: 25%

Filter: 12, 24, Clean, Freq: 18.5 kHz, Res

Pitch Env: 0.0%, 0%, 0 st

Time, Tone, Volume: 0%, 70%, 0.0 dB

Analog

Osc1: 0.0 dB, Shape, Octave, Semi, Detune, F1/F2, LP24, Freq: 22.0k, Reso: 10%

Noise: 0.0 dB, F1/F2, F1

Color: 682 Hz

Quick Routing, Vibrato, Keyboard, Unison, Glide

Pitch Bend: +2 st

Osc2: 0.0 dB, Shape, Octave, Semi, Detune, F1/F2, LP12, Freq: 22.0k, Reso: 10%

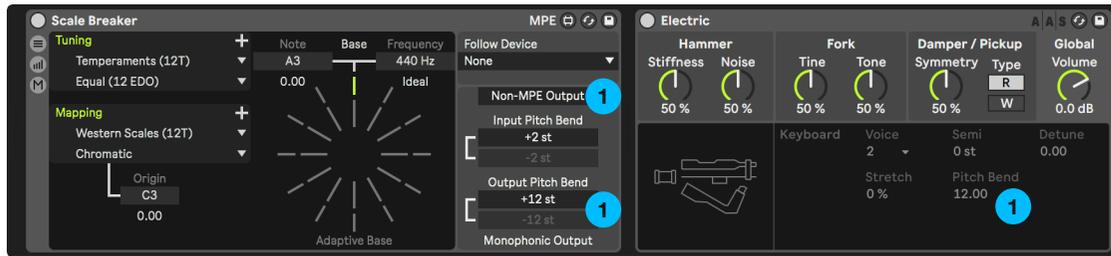
Follow, Amp1, Pan, Level: -18 dB

LFO1: Rate: 0.9 Hz, Vib: 0%, Rate: 5.1 Hz, Detune: 0.00, Uni: 0.00, Time: Legato

LFO2: Rate: 0.9 Hz, Gli: 50%

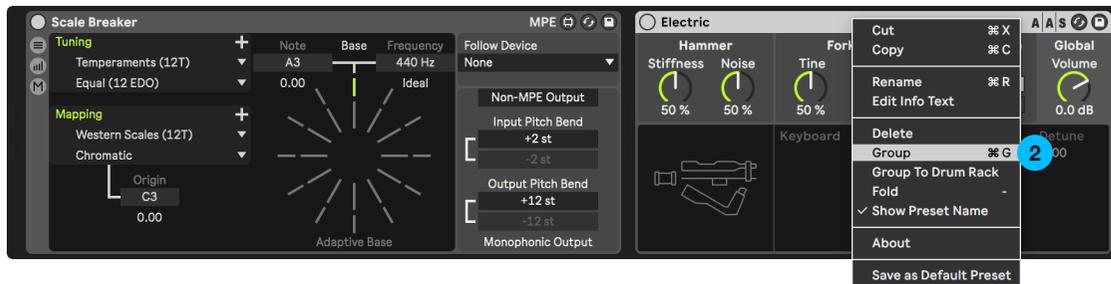


Setup: Connect to Non-MPE Instruments polyphonically



Regular instruments without **MPE** can only make use of the regular **Pitch Bend** that is applied to all notes. This makes it impossible to apply various tunings polyphonically.

A workaround on that is using multiple instances of the instrument. In that case, **Scale Breaker** will be able to feed each instance its own separate **MIDI** data.



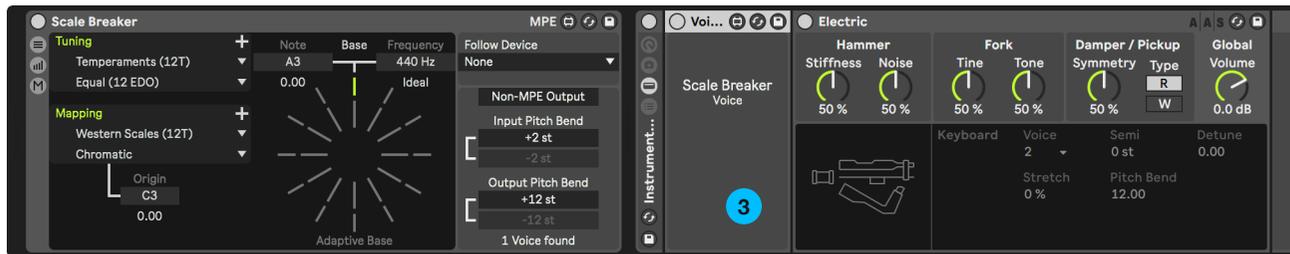
1 Set up the device with your non-**MPE** instrument, as described on the page **Connect to Non-MPE Instruments**.

2 Select your instrument and group it. The **Scale Breaker** device should not be a part of this group.

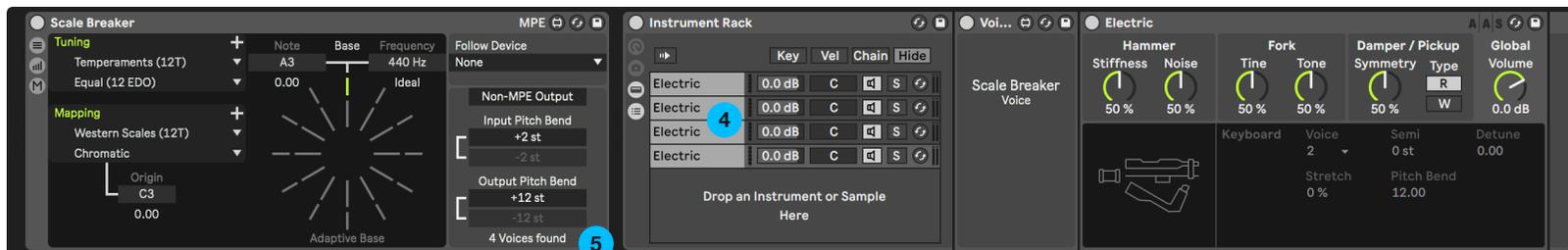
3 Insert the **Voice SB** device in front of the instrument within the group.

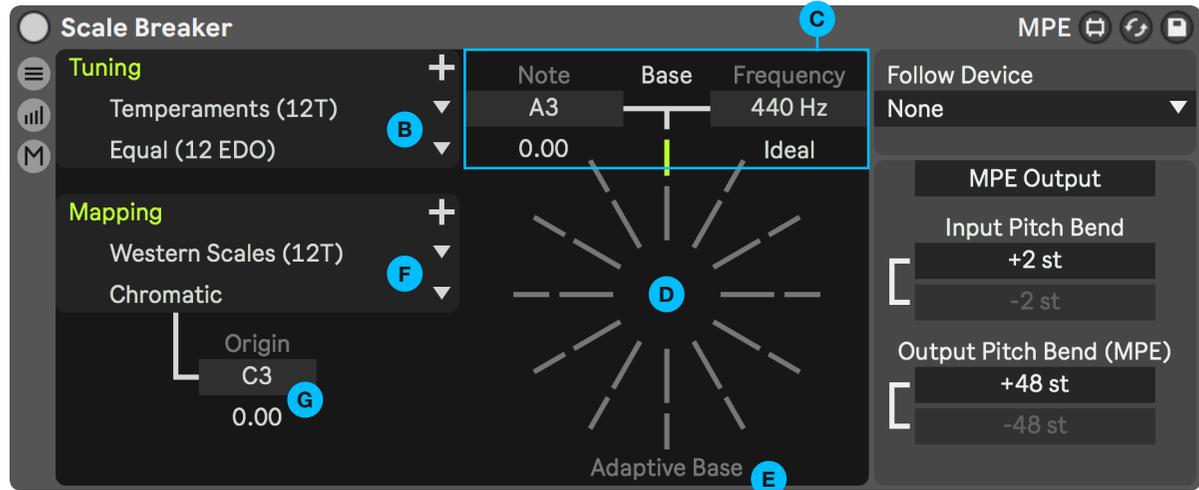
4 Open the chain list view of the group and duplicate the chain a few times. Every chain is one voice of polyphony. *Just as with MPE, you can have a maximum of 15 voices.*

5 The **Scale Breaker** device will locate the **Voice SB** devices automatically. It will then display, at the bottom right, how many voices have been found.

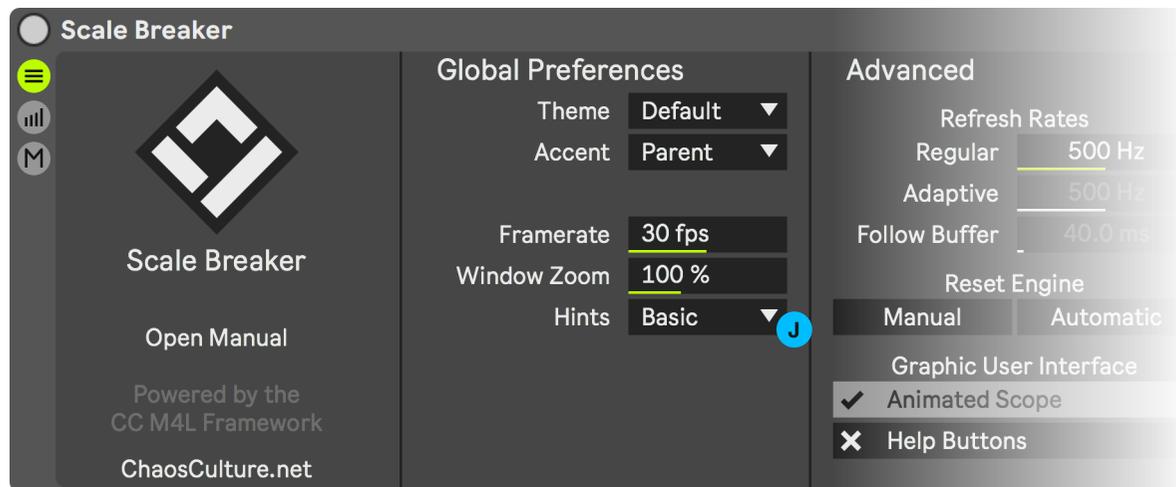


This only works if the **Voice SB** devices are positioned at the very front of the chain, and the group is located right after the **Scale Breaker** device.





- A** These buttons will open, respectively, **Preferences**, **Editor** and the **MPE Control** view.
- B** **Tuning** preset selection. The first menu chooses a preset group, and the second menu chooses a preset within it.
- C** This section controls the **Base Tuning**. The intervals of the loaded **Tuning** will be based on it. The **Ideal** button will set the frequency to the **12-TET** equivalent of the specified base note.
CMD/CTRL-Click: Quantize the Base Frequency to the BPM. Shift-Click: Quantize the BPM to the Base Frequency.
- D** The **Scope** in the center of the device provides a visual representation of the played notes. You can drag and drop **.scl** and **.kbn** files on top of it in order to quickly import them without having to open the **Tuning & Mapping Editor**.
- E** This toggle can activate the **Adaptive Base** system.
- F** **Mapping** preset selection. The first menu chooses a preset group, and the second menu chooses a preset within it.
- G** This **Origin Note** controls from which note the **Mapping** starts its sequence. If you're using one of the presets for quantizing a **12-Tone Tuning** into a **Scale**, this would specify its root note.
- H** This section is used to follow the tuning of other **Scale Breaker** devices in your **Liveset**. More information can be found on the page **Follow Device**.
- I** These controls are important to set up **Scale Breaker** to work correctly with your instrument. Please look at the pages **Connect to MPE Instruments** and **Connect to Non-MPE Instruments** for more information.
- J** Activate **Complete Hints** in **Preferences** to display additional tooltips when hovering over GUI elements. You can also use **Live's Info View** instead.





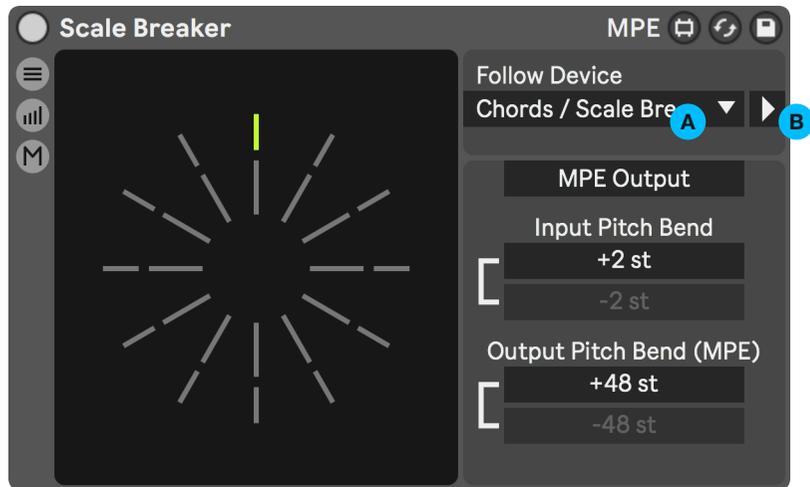
The **Adaptive Base** allows you to temporarily overwrite the **Base Tuning**, such as A3=440Hz, to any other note and its equivalent frequency, based on the currently used **Tuning** and **Mapping**. This is particularly useful for **Just Intonation** tunings.

- A** When activated, every note with a velocity of 1 will be filtered out and overwrite the **Base Tuning** instead. Within a **MIDI Clip** with some chords, you could duplicate all the root notes to an octave lower and set their velocities to 1, for example. None of those notes will reach the instrument; they'll alter the **Base Tuning** so it fits the played chords.
You don't have to use the root notes, sometimes using the 5th can be advantageous for short transitional chords.
- B** Only a single note of velocity 1 can be recognized at a time, but velocity 2 can be used as a second layer simultaneously. This second layer will apply a new **Base Tuning** relative to the first layer. By default, up to 4 layers can be used, meaning that all notes with velocities of 1 to 4 are designated for the **Adaptive Base** system, and will be filtered out. You can change the amount of layers with the **Layers** parameter.

Why use an Adaptive Base?

An **Adaptive Base**, or a similar system, is essential to make **Just Intonation** tunings even usable for most chord progressions. It's mathematically impossible to create a tuning where all notes have simple relationships to each other and therefore sound "in tune". Playing a Cmaj chord with a **Just Intonation** tuning is not going to sound great if it's based on A, since the fundamental frequencies for most notes do not have a simple relationship to C.

This is why **12-TET** is the standard tuning in modern western music. All the notes are logarithmically equally spaced apart, making them equally "out of tune" regardless of the used key or root notes. It's a compromise that allows musicians to play in any key without having to change the tuning for every chord, or use multiple, differently tuned instruments. This is also why using an **Adaptive Base** on a tuning of equal divisions has no effect.



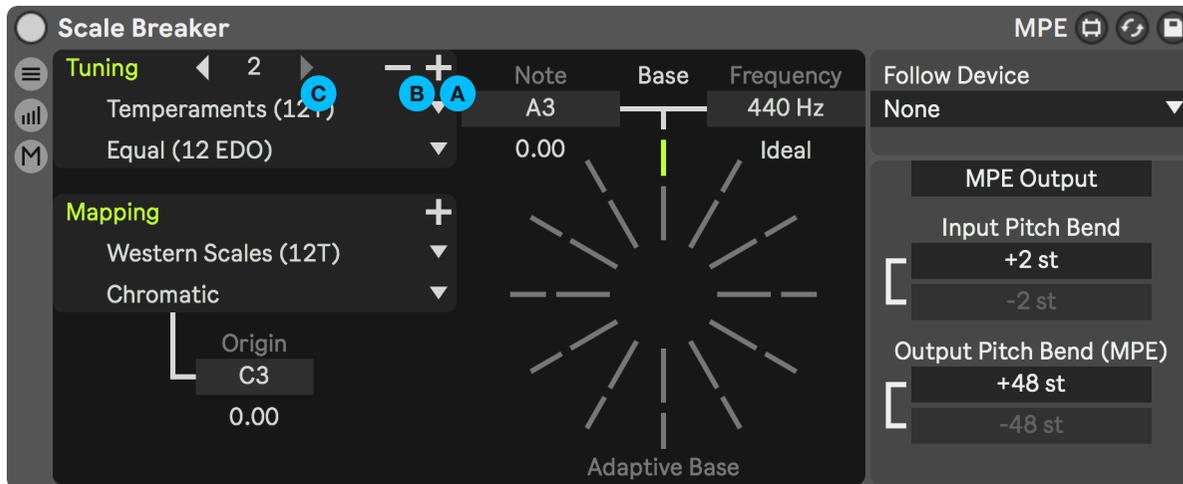
- A** You can follow the **Tuning** and **Mapping** of any other **Scale Breaker** device in your **Liveset** by selecting it in the **Follow Device** drop-down menu.

This way, you could have a single **Scale Breaker** device controlling the **Tuning** and **Mapping** of multiple instruments in your **Liveset**. As soon as you change the **Tuning**, all follower devices will be updated right away, including any notes that are currently being played.

Additionally, the **Adaptive Base** is also being followed. For instance, if you want to write a song in **Just Intonation**, you only need to layout the **Adaptive Base's MIDI** notes once, and every instrument can follow it. No need to manually copy and paste these notes to all other **MIDI Clips**.

- B** The arrow next to the drop-down menu will navigate your view in **Live** to the device that is being followed.

Make sure to insert an instrument after the followed device and have its output routed to the Master track. Even if the instrument never makes a sound, this is important for Live to recognize that the track it is on is used. Otherwise, Live will deactivate the processing of the track it's located on while rendering a song offline.



Instead of using one single **Tuning** or **Mapping**, it is possible to add up to 32 of them and use automation or **MPE Note Expressions** to select which one will be used at any given moment. For more information, go to the **MPE Control** page.

- A** The **+** button will duplicate the currently selected slot, adding it to the list.
- B** Use the **-** button to delete the currently selected slot. *Be aware that any slots after it will be shifted up, which might be unwanted if you're automating slot numbers.*
- C** The arrow keys allow you to browse through all available slots.



The screenshot shows a MIDI piano roll with notes G3, F3, D3, and E3. Below the piano roll are two tracks: Slide and Pressure. A blue box labeled 'A' highlights a note's MPE control data. A blue box labeled 'B' highlights the 'Slide' parameter for a note. A blue box labeled 'C' highlights the 'Mode' dropdown for a note. A blue box labeled 'D' highlights the 'Pressure' parameter for a note.

A In **Live 11**, it is possible to use **MPE Note Expressions**, such as **Slide** and **Pressure**, to change the **Tuning** and **Mapping**, or to use the **Freeze** functionality to create smooth transitions. Every note can be controlled individually.

B Select which **MPE Note Expression** input to use in order to control the specified aspects of the device. *The Custom CC option allows you to choose any control change number. Keep in mind that this is MPE control change, which can't be edited in Live's MIDI Editor.*

C Modes for controlling **Tuning and Mapping Slots**:

- **Default**: The control value (0 to 127) is used to specify the slot that will be used. A 0 will bypass this **MPE** control, and the automatable parameter will be used instead.
- **XXO**: The last digit of the value will be ignored. That means a value of 118 will use slot 11 and a value of 62 will use slot 6. This is useful in combination with the next option.
- **OOX**: Only the last digit will be used, so values of 37 or 127 will both use slot 7.

With these two non-default options, it becomes possible to individually control the **Tuning** and the **Mapping** slots from the same **MPE Note Expression**.

D Modes for controlling the **Freeze**:

Default simply uses the MPE note expression value (0 to 127) for the freeze amount. Additionally you can invert this value with the option **Invert**, which might make it easier to write music.

E All of these MPE control routings can be set up for notes that are used with the **Adaptive Base** system. Everything works in the same way. You can change the **Tuning** or **Mapping** slots, or also freeze them, to create smooth transitions between different base tunings.

Adaptive base notes are separated from the regular notes here, so it can be possible, for instance, to use the MPE slide to affect the Adaptive Base while still having the MPE slide data of regular notes being sent to the instrument.

The screenshot shows the 'MPE Control' settings panel. It has two sections: 'Regular Notes' and 'Adaptive Base'. Each section has three rows: 'Tuning Slot', 'Mapping Slot', and 'Freeze'. Each row has a 'Source' dropdown and a 'Mode' dropdown. Callouts B, C, D, and E point to these dropdowns in the 'Regular Notes' section.

| Section | Parameter | Source | Mode | |
|---------------|---------------|--------------|---------|---------|
| Regular Notes | Tuning Slot | Slide | Default | |
| | Mapping Slot | Off | Default | |
| | Freeze | Pressure | Default | |
| | Adaptive Base | Tuning Slot | Off | Default |
| | | Mapping Slot | Off | Default |
| | | Freeze | Off | Default |

Tuning

Just Intonation (12T)

5-limit Symmetric 1 (au)

Mapping

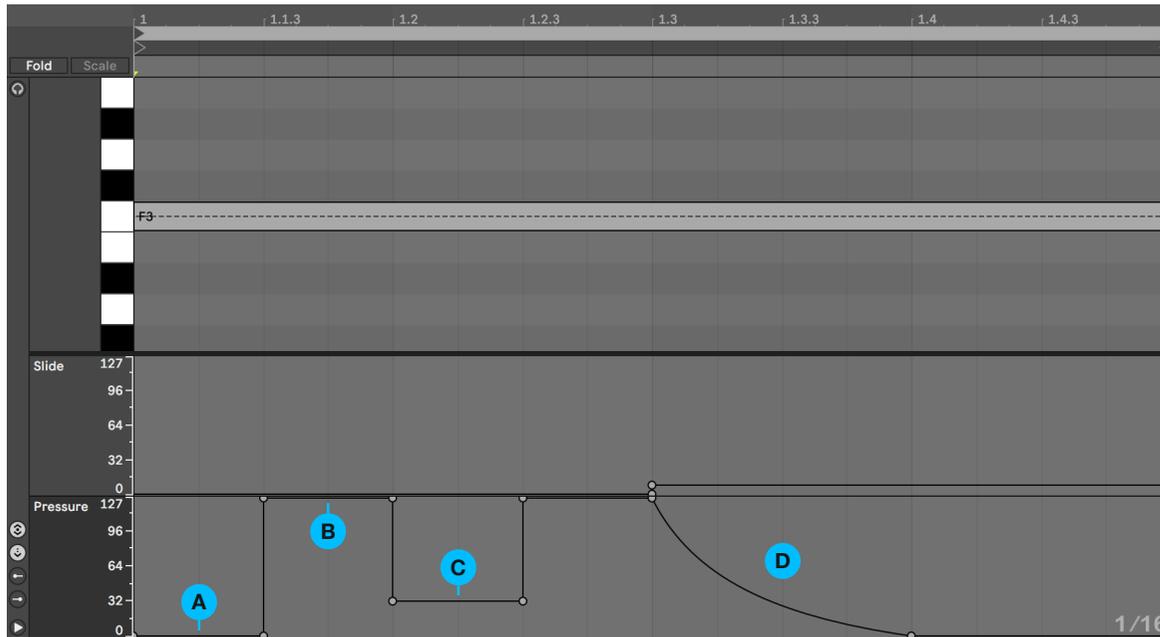
Western Scales (12T)

Chromatic

Origin Layer

C3 4

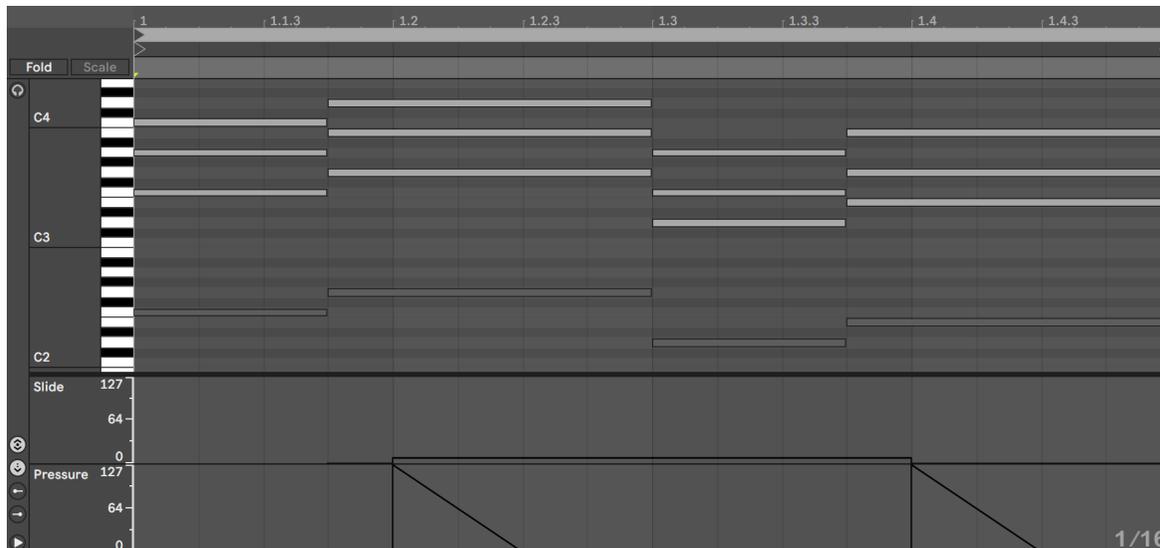
0.00

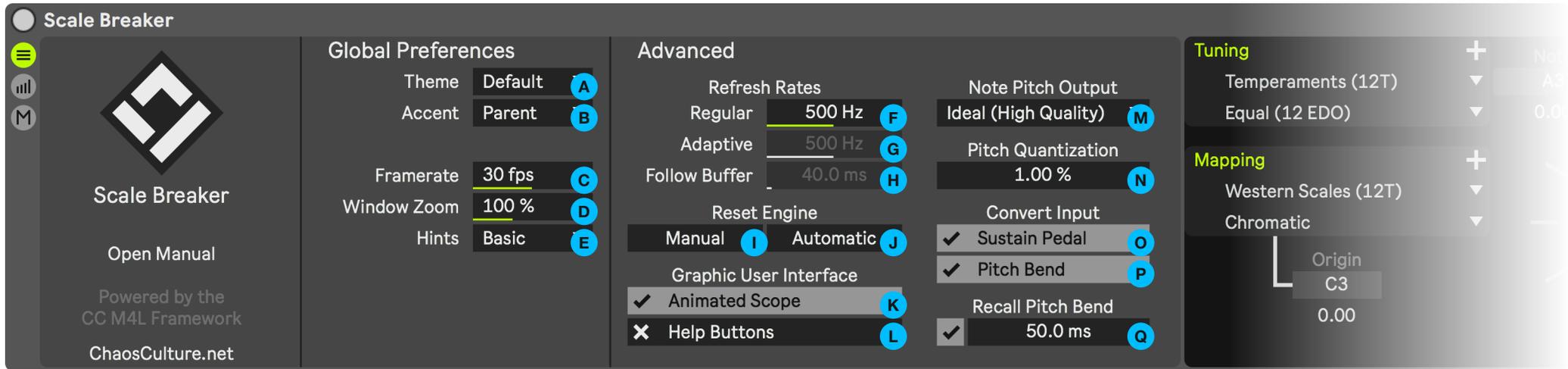


Freezing notes is an advanced feature that can be used to create smooth transitions between any frequency changes. It is only possible by using **MPE Note Expressions**, and therefore not available for **Live** versions below 11. You have to route an **MPE Note Expression** input to the **Freeze** functionality in order to use it. Take a look at the page **MPE Control** in order to find out how to do it.

Every note has a **Freeze** amount, which is a value between 0 and 127. It acts similar to a **Sample & Hold** for the frequency. *The frequency is not technically being frozen, but the ratio of a note relative to the Base Frequency is, so that it is still possible to change it even if some notes are frozen.*

- A** When the **Freeze** value is set to 0, the note plays normally and its frequency is sampled for future use.
- B** When the **Freeze** value is set to 127, the note is frozen and will use the previously sampled frequency.
- C** Any number between 0 and 127 is used to interpolate between the sampled frequency and the unfrozen note.
- D** The moment a note changes its frequency, you can insert a shape going from 127 to 0, as shown in the example, to create a smooth transition. This allows you to draw any transition shape and duration individually for every note.
- E** Notes that are used by the **Adaptive Base** system can also be frozen. This way, you can create smooth transitions between different base tunings. *When using multiple layers, only the highest Freeze amount that is played at any moment will be used.*





- A** This list contains all available **Color Themes**. Every Theme is an algorithm that reacts to the Theme used in **Live** in different ways.
- B** This list contains all available sources that can be used as the accent color for the selected Theme.
- C** Animated elements, other than parameters, are tied to this refresh rate. A lower frame rate can relieve stress on **Max**, but the device will also automatically drop frames when necessary.
- D** All external windows will be scaled by this amount. It is the equivalent to the **Display Zoom** in **Live**'s preferences.
- E** Almost all GUI elements can display hints when hovering over them with the mouse cursor. This menu allows you to choose between different hint options.
- F** Determines how frequently the **Pitch Bend** of a note can be updated. Higher frequencies result in better resolution and shorter latency times, whereas lower frequencies are more efficient if the device has to process a lot of **MIDI** data in a short amount of time.
- G** Determines how frequently the **Adaptive Base** system can be updated. Higher frequencies result in better resolution and shorter latency times, whereas lower frequencies are more efficient if the device has to process a lot of **MIDI** data in a short amount of time.
This is not used if this device is following the tuning of a different one, or if the Adaptive Base system is deactivated. However, other devices that are following this one will be affected by this setting.
- H** This value is only relevant if this device is following the **Tuning** and **Mapping** of another device. It determines the maximum amount of delay time that can be compensated due to communication delays across devices.
This parameter will flash up whenever the delay compensation was not able to compensate a message correctly. Increasing this value will help reducing these situations.
- I** Resets the engine, which can be useful if notes get stuck for some reason.
- J** Whenever the playback is stopped, the engine will reset itself automatically if this toggle is activated.
This only includes things that can be reset efficiently. The manual reset button is able to reset more things.
- K** Shows or hides the animated **Scope** display. Hiding the **Scope** results in a more efficient GUI. This setting is synchronized across devices.
- L** Shows or hides **Help Buttons** in various areas. This setting is synchronized across devices.
- M**
 - **Original**: Every note will keep its original pitch and only the **Pitch Bend** data is used to offset them.
 - **Ideal**: This setting allows the device to freely choose which notes to output in order to be as close to the desired frequency as possible. This can be crucial for certain tuning systems, as they would otherwise require large **Pitch Bend** output ranges.
 - **Ideal (High Quality)**: This mode will additionally avoid the clashing of notes due to their frequencies being too close together. Some notes could otherwise get lost in some tuning systems.
- N** The incoming **Pitch Bend** data is rounded to semitones to compensate for inaccuracies due to **MIDI** limitations. 1% is exactly the right amount to solve these issues. 100% will quantize the pitch bend completely, making it impossible to play in-between notes. This can also be used to make it easier to hit specific notes on an **MPE MIDI Keyboard**.
- O** With this toggle activated, the **Sustain Pedal** input will be converted, causing the notes to play for longer. This can be useful, as it's not possible to change the frequency of notes that are only being held by the **Sustain Pedal**.
- P** By deactivating this toggle, you can bypass regular **Pitch Bend** inputs and pass them through to the instrument without translation. This is only possible with **MPE Instruments**.
- Q** Recalling the **Pitch Bend** outputs will prevent **Live** from smoothing the **Pitch Bend** data and interpolating it in places that are not desired. Whenever a **Pitch Bend** hasn't moved for the specified amount of time, the device will recall the previous **Pitch Bend** value on the next change to ensure that it is not going to be turned into a ramp up or down by **Live**.



This Editor allows you to create custom **Tunings** and **Mappings**.

- A** To open the **Editor** window, use its button at the very left of the device, or select **Custom** within the preset drop-down menus.
- B** You can select which slots to edit. See **Tuning & Mapping Slots** to learn more about them.
- C** The presets are also available from this window. Editing the content of a preset will automatically turn it into a custom one.
This makes it possible to use presets as starting points for your own Tunings or Mappings.
- D** The **Intervals**, **Sequence** and **Formal Octave** fields are text inputs describing a **Tuning** or a **Mapping**. The next pages will explain which values can be used and how they are interpreted by the device. *You can copy and paste text within these inputs using the regular shortcuts of your OS.*



Tuning +

Preset

Temperaments (12T) ▼

Equal (12 EDO) ▼

Intervals

0

100

200

300

400

500

600

700 A

800

900

1000

1100

Formal Octave

1200 B

Scale File (.scl) Import Export

A **Tuning** consists of a list of intervals, which is repeated indefinitely with an offset for every repetition that is set by the **Formal Octave**.

- A** The list of **Intervals** starts with the tonic; every line of text below it describes the following intervals. In a traditional 12-tone system, this would be the minor second, major second, minor third, and so on. There is no limit to how many intervals there can be in one octave, add as many lines of values as you wish. There are multiple ways to describe an interval:

A number, integer as well as decimal, will be recognized as a cents value. 100 cents is one semitone of the standard 12 EDO tuning.

Examples: **100 100. 100.52 367.3182204**

A ratio can be written using two numbers separated by a forward slash. These are typically used for **Just Intonation** tunings. It is possible to use decimal numbers here as well. This might seem counterintuitive, but it can be useful to add a slight detuning, or to give more flexibility when experimenting with weird intervals.

Examples: **4/3 16/15 3.37/8.224**

For tuning systems using equal divisions of an octave (EDO or TET), you can write the step number, add a space, and then write the EDO system name in one word afterwards.

Examples: **4 12EDO 17 24EDO 5.2 13EDO**

Furthermore, you can use equal divisions of any interval. Instead of typing **EDO**, you can also use **EDF, EDV, EDX, EDT, EDP, EDS, EDN**, or any interval as a number or ratio after **ED**.

Examples: **13 32EDX 7 10EDV 22 31ED7 6 12ED5/4 7.31 35.8ED10.73/5.28**

All these methods are different ways to calculate a value that will be used to determine the frequency of a note by multiplying it with the **Base Frequency**. If you want to calculate this value yourself, you can directly type it in by adding an equal sign in front of it.

Examples: **= 1.5 = 1.33333333 = 0.285**

- B** The **Formal Octave** is also an interval, so it can be described using the same rules. It might be helpful to learn the different ways of describing intervals by looking at the included preset tunings.

Please note: Tuning files (.scl) do not support detuning of the tonic. Exporting will not be possible in this case. Some intervals will also be converted to cents when exporting.



A **Mapping** is used to change the layout of the **MIDI Keyboard** by reassigning the keys. This can be useful to make **unconventional tunings** easier to play on a regular 12-tone keyboard, to create subsets of a **Tuning** or to quantize the notes into a musical scale as a lot of the included presets do.

- A** The **Sequence** starts at the **Mapping Origin**, which is a note specified with an automatable parameter shown at **B**. In most cases, it's suggested to keep it at 0 so that the origin note will not be changed.

Each following line is a value that specifies an offset in pitch relative to the origin that should be played. The second number will be used as the pitch that is located one note above the origin. A 1 will therefore yield no change. By entering a 5 instead, playing the note above the origin will actually end up playing 5 notes above it, but which interval this ends up being depends on the **Tuning**.

This mapping system also allows you to use decimal numbers, making it possible to play notes in between intervals regardless of the **Tuning**.

- C** The **Sequence** can have any size, so you may add as many lines of values as needed. It will be repeated automatically with an offset specified by the **Formal Octave**.

Please note: Keyboard Mapping files (.kbm) do not support decimal numbers. Exporting will not be possible in this case.