ODDSOUND : MTS-ESP MIDI CLIENT

User Guide
Version 1.11 (31 Oct 2022)

Oli Cash
Dave Gamble
Damon Hancock

MTS-ESP MIDI CLIENT: TABLE OF CONTENTS

Windows

- MTS-ESP MIDI Client works with Windows 7, 8, and 10, 32bit and 64bit.
- VST, VST3 and AAX are supported in both 32bit and 64bit hosts.
- VST MIDI FX is offered for Cubase and Nuendo (32bit and 64bit) as an experimental feature.
- Your graphics card must support OpenGL2. This should include most cards, though performance with integrated Intel graphics may be poor.

Mac OS

- MTS-ESP MIDI Client requires OS X 10.7 or newer on Intel or ARM Mac.
- VST, VST3, AU and AAX are supported in 64bit hosts only.
- VST MIDI FX is offered for Cubase and Nuendo as an experimental feature.

How To Install On Windows

You need two files to install, the ODDSound MTS-ESP Suite installer and a licence file. Both are available from the Downloads page at **https://oddsound.com** whilst logged in. Note that there are separate 32-bit and 64-bit installers for Windows.

For a 30-day demo, click 'Demo' on the Downloads page to request a demo licence file to be emailed to you.

To install the license, double-click the license.reg file and this box will open:
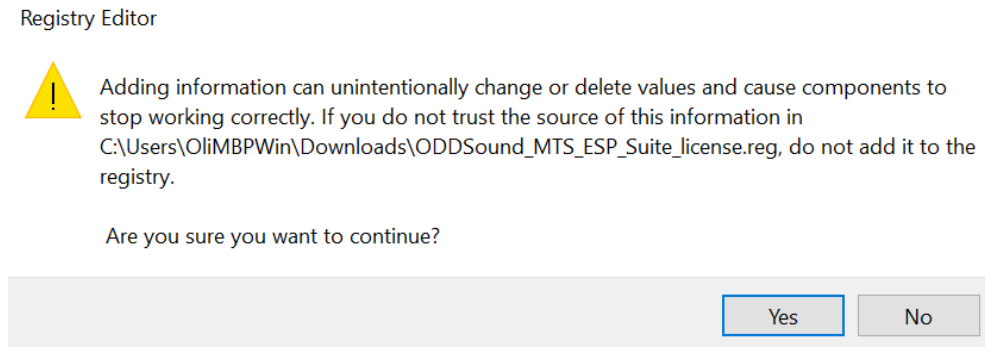
Click 'Yes' and the license will be installed.

Next unzip and double-click the installer to run it. It will look like this:
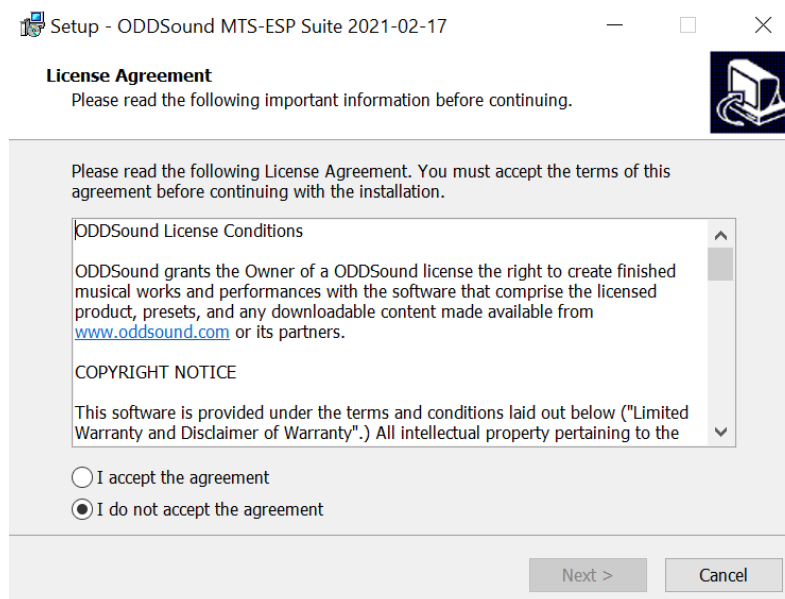
Accept the agreement and click 'Next' to get to the installation options:
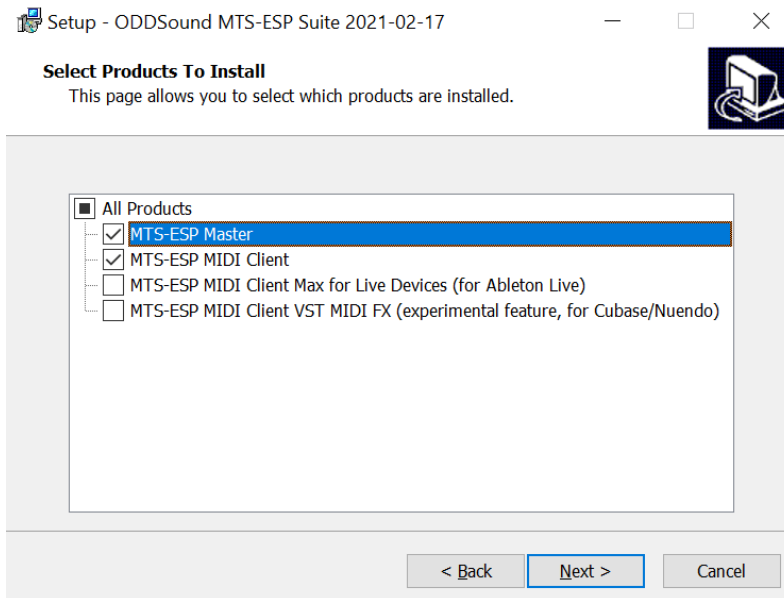
Select the components to be installed and click 'Next' to choose which plug-in formats to install:
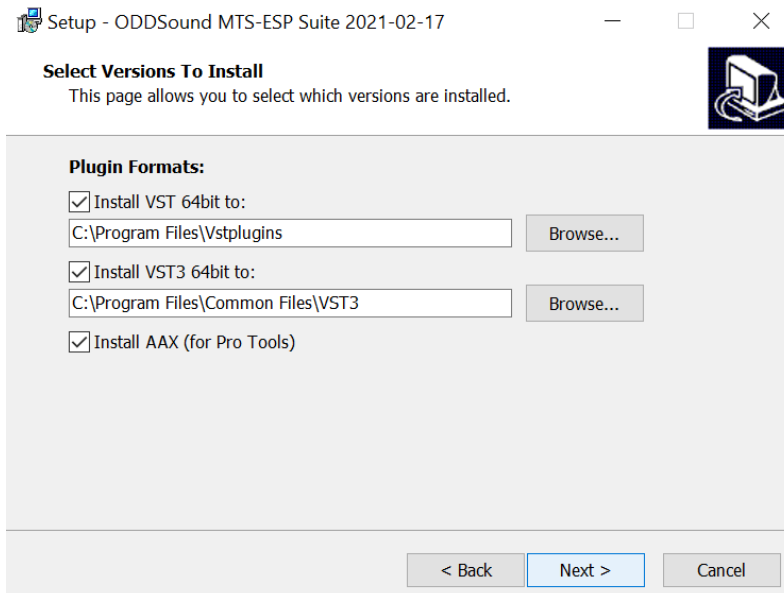
Enable the check boxes for the plug-ins formats to be installed. The VST location should be set to your VST Plugins folder. The usual location for VST3 plug-ins is \Program Files\Common Files\VST3 but this can be changed if required.

If installing the MTS-ESP MIDI Client Max for Live devices, you will be prompted to select the location to install them to:
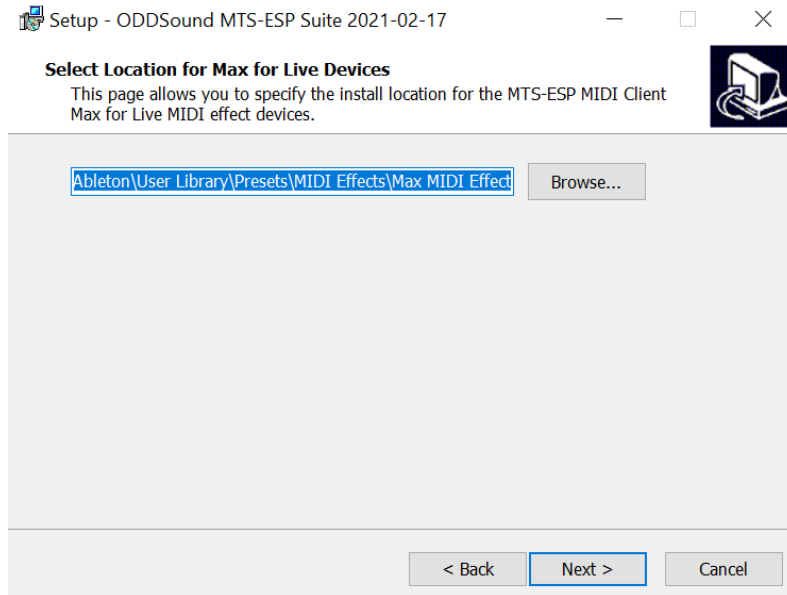
The default location is \Users\[username]\Documents\Ableton\User Library\Presets\MIDI Effects\Max MIDI Effect. If the Live User Library is located elsewhere you will need to edit the install location.

Note that, unless it was previously installed, installing the MTS-ESP MIDI Client Max for Live devices will block the VST3 version of MTS-ESP MIDI Client from being installed, to prevent Max attempting to load it. The MTS-ESP MIDI Client Max for Live devices require the VST2 version to work correctly.

Once all options have been set, click 'Next' and 'Install' to install everything and then start/re-start your DAW to begin playing!

How To Install On OSX

You need two files to install, the ODDSound MTS-ESP Suite installer and a licence file. Both are available from the Downloads page at **https://oddsound.com** whilst logged in.

For a 30-day demo, click 'Demo' on the Downloads page to request a demo licence file to be emailed to you.

If using Safari, one or both of these files may be automatically unzipped after downloading. If not, double-click to decompress them so you get this:
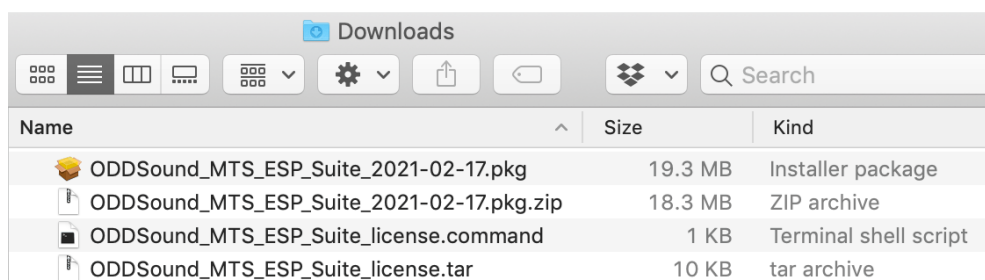
Right-click the "license.command" file and select 'Open' to install the license. The license will install and you'll get the following message to confirm:
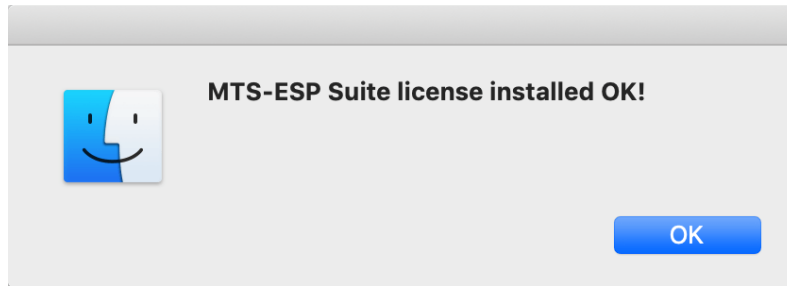
Right-click the installer .pkg file and select 'Open' and continue through installation to the 'Installation Type' page:
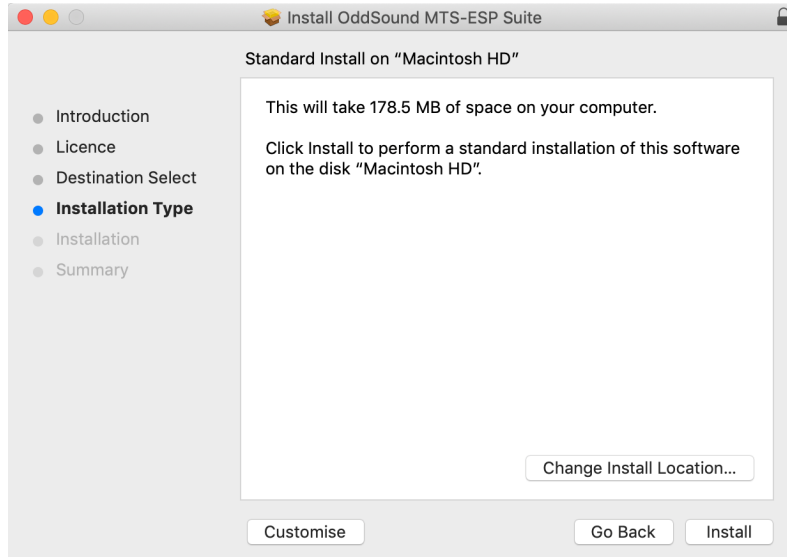
To select the components to install click 'Customise':
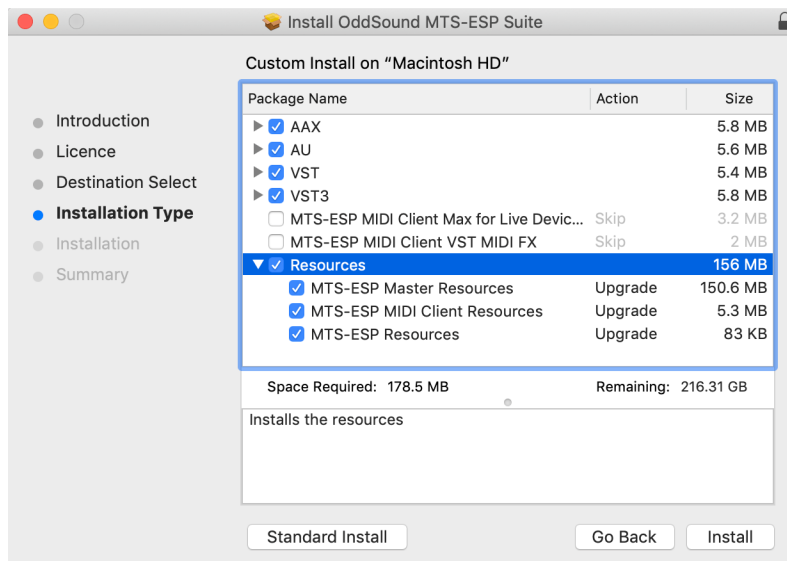
Note that if installing the MTS-ESP MIDI Client Max for Live devices, these must be installed into the Live User Library, the default location for which is /Users/[username]/Music/Ableton/User Library. If the User Library location is different from this, manually copy the Max for Live devices, plus the accompanying "MTS-ESP MIDI Client.vst" file, to the Live User Library after running the installer. Copy them from:

/Library/Application Support/ODDSound/MTS-ESP MIDI Client/Max for Live MIDI Effects

to the following folder inside the Live User Library:

User Library/Presets/MIDI Effects/Max MIDI Effect

MTS-ESP MIDI Client offers a way for hardware devices, or plug-ins which do not have MTS-ESP support, to automatically follow the tuning from the MTS-ESP Master plug-in.

It is a MIDI plug-in, meaning it accepts and outputs MIDI data. It does not process or output any audio. It is intended to sit between a source of MIDI data and its destination, adjusting or generating MIDI messages as necessary for retuning.

Unfortunately there is no well agreed standard for MIDI plug-ins, so in a DAW it will usually be inserted as an instrument plug-in and MIDI data must be routed from it to a target plug-in or output MIDI port. That said, some DAWs do offer better support for MIDI plug-ins. Instructions for how to set up MTS-ESP MIDI Client in specific DAWs can be found at the end of this manual.

There are three core ways in which retuning can be achieved by MTS-ESP MIDI Client:

MIDI Pitch Bend

This method uses MIDI pitch bend messages to retune a synth as it is playing. It has the advantage that it will work with any device or plug-in that responds to pitch bend messages. There are however limitations, for which MTS-ESP MIDI Client offers workarounds:

- The pitch bend range of the synth limits the amount by which a note can be retuned. To get around this MTS-ESP MIDI Client has the option to transpose the played note to that nearest the target pitch before applying any necessary pitch bend.
- It is not usually possible to apply different pitch bend amounts to multiple notes at the same time, therefore this method suits monophonic parts. For polyphonic parts, some synths can be configured to have a 'voice-per-MIDI channel', allowing each note and its corresponding pitch bend messages to use a separate MIDI channel. Synths that support MPE (MIDI polyphonic expression) work this way too. Multi-channel and MPE poly modes are available in MTS-ESP MIDI Client for synths that support them.
- The pitch bend control on a synth can no longer be used with this method. Glide may also not work as expected with the retuning, so pitch bend and glide controls are offered in MTS-ESP MIDI Client to replicate these features using MIDI pitch bend messages. Note that pitch bend should still work correctly with synths that support MPE, as global pitch bend can be sent via the master channel.

MTS SysEx

MTS (MIDI Tuning Standard) is a specification of precise musical pitch agreed to by the MIDI Manufacturers Association in the MIDI protocol. MTS uses MIDI SysEx messages to transfer tuning information between MIDI devices. Some hardware synths offer MTS support. Refer to the manual for a specific synth to see if it supports MTS and, if so, which MTS message format(s) it will accept. MTS-ESP MIDI Client supports all MTS message formats and includes a number of presets for hardware synths. Full details of the message formats can be found at the MIDI Manufacturers Association website:

https://www.midi.org/specifications-old/item/the-midi-1-0-specification

VST Detune

When MIDI messages are passed between a plug-in and its host, there is usually some extra data accompanying the MIDI message. In the VST2 and VST3 specifications, this includes a 'detune' parameter, which allows for per-note retuning.

Advantages to using this method are that the pitch bend messages are unaffected and it works polyphonically without using multiple MIDI channels.

However, detune is applied only at note-on events, so real-time changes to tuning will not take effect whilst a note is held. Also it is not guaranteed to be supported by a host or plug-in and some ignore it. It will not work at all with AU or AAX plug-ins. Further details are given in the 'Method: VST Detune' section.

Proprietary Methods

Some synths or virtual instruments may use a proprietary method for retuning that allows real-time updating. Synths2 currently supported include the Yamaha FB-01 and Waldorf Wave and Microwave I. Support for others may be added in future updates.

Bypass

Undo/Redo   Panic

ODDSOUND   MTS-ESP MIDI CLIENT   ↺ C PRESET ! ⏻

Undo/Redo

A full undo/redo for any parameter changes on the UI, including preset selection.

Preset

Click for preset load/save options:

- **Load preset…:** Load a .MIDIClientPreset file.
- **Save preset…:** Save a preset as a .MIDIClientPreset file. Preset files saved in the following locations will appear in the preset menu for quick selection:

  Win: Documents\OddSound\MTS-ESP MIDI Client\Presets

  OSX: Library/Audio/Presets/OddSound/MTS-ESP MIDI Client
      ~/Library/Audio/Presets/OddSound/MTS-ESP MIDI Client

- **Save as default:** Save the current state of MTS-ESP MIDI Client as default, which is loaded on instancing.
- **Reset to default:** Reset MTS-ESP MIDI Client to the saved default state.
- **Default preset folder (OSX only):** OSX offers two locations for saving plug-in presets: 'local', accessible by all users, and 'user', only accessible by the current logged in user:

  local: Library/Audio/Presets/

  user: ~/Library/Audio/Presets

  This option selects which of the above folders MTS-ESP MIDI Client defaults to when saving presets. Presets saved into the selected folder will appear in the top level of the Preset menu, whilst presets saved into the other will appear in a sub-menu.

  Factory presets are stored in the local preset folder. Factory presets will be overwritten whenever installing an update to MTS-ESP MIDI Client. If any edits are made to a factory preset, it should be stored either with a new name or in the user presets folder.

Panic Menu

This menu contains options that may be useful when working with hardware devices:

- **Reset pitch bend:** This will send a pitch bend message with value 0 (center) to all MIDI channels currently in use. If the tuning of a synth sounds incorrect, pause playback and try this option.
- **Send all notes off:** This will send an all notes off MIDI message (CC 123, value 0). When using MTS SysEx, a hardware synth may end up with stuck notes if it can't cope with the volume of SysEx messages it is receiving. If

this happens, use this option and then try increasing the SysEx Throttle value.

- **Send 12-TET SysEx dump:** When using MTS SysEx, or Waldorf Wave proprietary retuning, this option provides a quick way to return a device to its original 12-TET tuning.

Bypass

When engaged, **Bypass** will reset the tuning to 12-TET. The tuning in MTS-ESP Master will be ignored. Right-click for MIDI CC learn options. MIDI CC assignment is global and will apply to all instances in all sessions. The assigned MIDI CC will be filtered from the plug-in MIDI output on all MIDI channels.

Controls

Controls and parameters that respond to mouse actions are coloured light blue.

For numeric value parameters, you can click and drag, use the mousewheel or double-click and enter a number to set the parameter value. Hold shift whilst dragging or mousewheeling for fine adjustment.

Hold Ctrl/Cmd whilst clicking a parameter to reset it to its default value.

Connection Info

This shows the state of the connection to MTS-ESP Master, either **Connected**, **Disconnected**, or **Connected (Master Bypassed)**, in which case the tuning will be reset to 12-TET and the controls in the MTS-ESP MIDI Client will have no effect. Connection between MTS-ESP MIDI Client and MTS-ESP Master is automatic.

Retune Method Menu

This menu allows selection of the retuning method. See the 'Overview Of Use' section for an overview of the methods available. The controls available on the UI will change depending on which method is chosen and are described in the following sections.

Mouseover Help

A mouseover tooltip is provided at the bottom of the UI. Hover the mouse over a parameter to see information about its function.
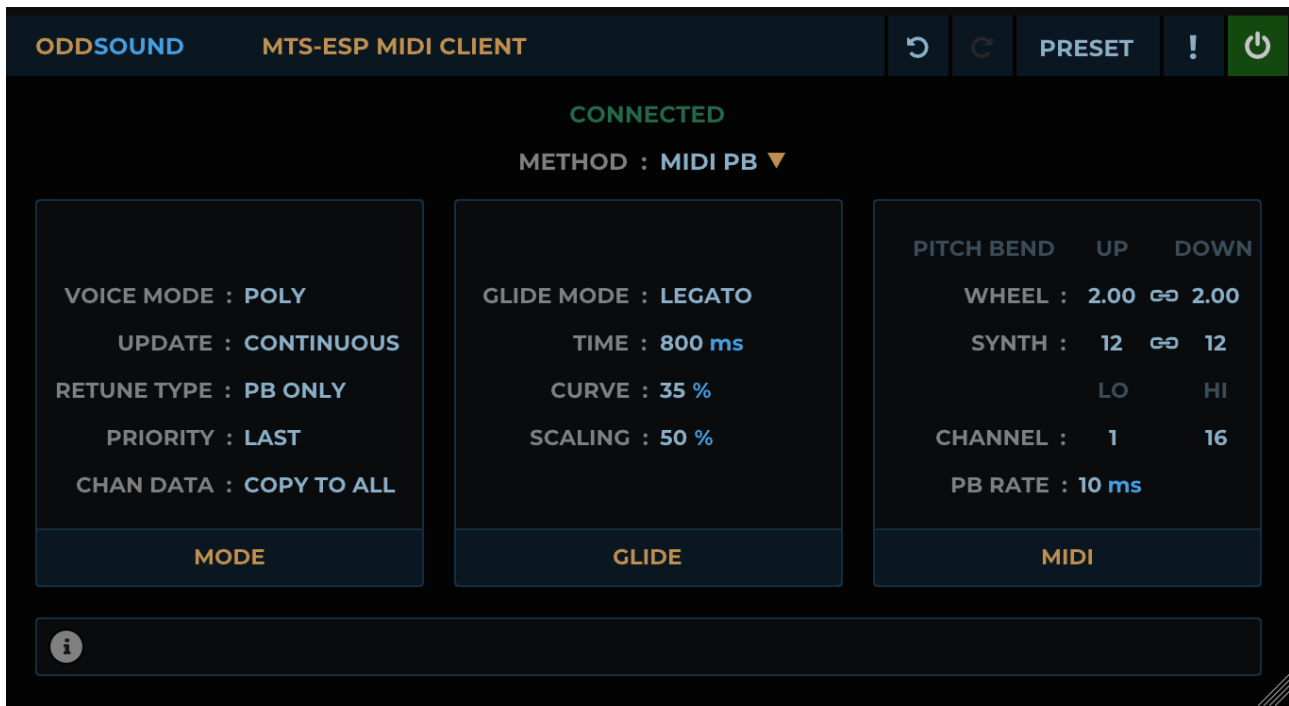
This method uses MIDI pitch bend messages to retune a synth as it is playing. It is best suited to monophonic parts, however if a plug-in or hardware synth supports 'voice-per-MIDI channel' or MPE, it can be used for polyphonic parts too.

Mode Parameters

Voice Mode:

- **Mono:** Note and pitch bend data are sent on a single midi channel. This mode is suited to monophonic parts.
- **Poly:** Note and pitch bend data will be sent over a range of MIDI channels. Use this mode for polyphonic parts with synths that support voice-per-MIDI channel.
- **Poly MPE:** Note and pitch bend data will be sent over a range of MIDI channels in accordance with the MPE specification. Use this mode for polyphonic parts with synths that support MPE. Note that you may need to explicitly enable MPE in the synth being retuned.

Update:

- **Continuous:** Retuning is updated continuously, whilst notes are held.
- **Note-On Only:** Retuning is only applied at note-on. The pitch of a note will not be updated whilst it is held.

Retune Type:

- **PB Only:** Retuning is achieved by using pitch bend messages to adjust the pitch of the played note.
- **Note+PB:** Retuning is achieved by first transposing the note to that nearest the target pitch, then using pitch bend to get to the exact pitch. This is useful if the synth being retuned has a small pitch bend range, or for achieving greater retuning accuracy. Note that the transposition only happens with a note-on event. The pitch bend range of the synth will limit the amount by which the tuning of a note can be changed whilst it is held.

Priority: This parameter functions differently depending on whether **Voice Mode (page 15)** is set to 'Mono' or one of the poly modes.

In Mono mode, if multiple keys are pressed this parameter determines which of the held notes is used to calculate the tuning adjustment. If retuning a monosynth, set this parameter to match its priority.

In Poly or Poly MPE mode, this parameter determines what happens when the number of keys pressed exceeds the number of available MIDI channels as follows:

- **Last:** Turn off the first pressed key.
- **First:** Key press will be ignored.
- **Low:** Turn off the highest note or, if the key is higher than all those currently held, ignore key press.
- **High:** Turn off the lowest note or, if the key is lower than all those currently held, ignore key press.

This parameter is not available for Poly MPE mode if **MPE Receive (page 17)** is on, as it is assumed notes are already distributed among MIDI channels.

Chan Data: This parameter is only available if **Voice Mode (page 15)** is set to 'Poly' and determines what happens to channel messages, i.e. note, pitch bend, program change, control change, channel aftertouch and poly aftertouch:

- **Pass Thru:** Note messages are distributed among MIDI channels, along with their corresponding pitch bend for retuning. Program change, control change, channel aftertouch and poly aftertouch messages are passed through MTS-ESP MIDI Client with no change, they remain on the channel on which they were sent. This option would typically be used with an instrument that supports voice-per-channel.
- **Copy To All:** Note messages are distributed among MIDI channels, along with their corresponding pitch bend for retuning. Program change, control change and channel aftertouch messages received by MTS-ESP MIDI Client on any channel are copied to and sent out on all channels in the channel range selected in the **MIDI Parameters (page 17)** section. Poly aftertouch messages are passed through MTS-ESP MIDI Client with no change. This option would typically be used with a multi-timbral synth where the same sound has been loaded into multiple parts, each assigned to a different MIDI channel.
- **Multi Mono:** All channel messages, including note and pitch bend messages, remain on the channel on which they were received. MTS-ESP MIDI Client performs no channel reallocation of note messages. It may be seen as equivalent to having a separate instance of MTS-ESP MIDI Client for each channel with **Voice Mode (page 15)** set to 'Mono' and **Force Mono (page 16)** set to 'Off', hence the name 'Multi Mono'. This option would typically be used with a MIDI guitar controller where each string is assigned to a different MIDI channel.

MPE Voices: This parameter is only available if **Voice Mode (page 15)** is set to 'Poly MPE' and determines the number of MIDI channels that are used for MPE. If **Master Chan (page 17)** is set to 1 the channels used will extend upward from channel 2, else if set to 16 they will extend downward from 15. If **MPE Receive (page 17)** is on, any data received on channels not specified for use with MPE will be passed through the plug-in.

Force Mono: This parameter is only available when **Voice Mode (page 15)** is set to 'Mono'. If set to 'On', a note-off message will precede every note-on message to turn the previous note off, forcing a synth to play monophonically. If a synth has its own mono mode and it is enabled, set this to 'Off', otherwise it is advised to leave it 'On'.

Glide Parameters

If using MIDI pitch bend messages to retune a synth, the glide on the synth may no longer work as expected. The glide parameters in MTS-ESP MIDI Client emulate glide using MIDI pitch bend messages.

Glide Mode:

- **Off:** No glide is applied.
- **Legato:** Glide is triggered only when notes overlap.
- **Always:** Glide is triggered on every note-on.

Time: The time in milliseconds that it takes to glide from one note to the next.

Curve: Sets the curve shape that the pitch follows when gliding from one note to the next. 0% is linear, positive values are increasingly exponential and negative values are increasingly logarithmic.

Scaling: Sets the amount by which the glide time is scaled relative to the interval between the played notes, centered around an octave jump.

MIDI Parameters

Pitch Bend:

- **Wheel:** This applies when **Voice Mode (page 15)** is set to 'Mono' or 'Poly'. With these modes for retuning, the pitch bend control on the synth can no longer be used.

  'Wheel' pitch bend provides a way for incoming MIDI pitch bend messages to still have an effect on the synth, by adjusting the pitch bend messages according to the retuning. These values set the range in semitones for incoming pitch bend messages.

  They can be set to precise decimal values to allow for any interval and, if entered as text, integer ratios typed as two numbers separated by a '/' are accepted, e.g. "3/2". Alt-dragging on the range values will snap them to interval values from MTS-ESP Master.

- **MPE Bend:** This applies when **Voice Mode (page 15)** is set to 'Poly MPE' and **MPE Receive (page 17)** is on, which should be the case when the plug-in is receiving MPE data. Set the range to match the MPE bend range of the controller. This will usually be +/-48 semitones, although some controllers may offer an option to change this.

- **Synth:** This range MUST be set to match the pitch bend range on the synth being retuned, so that outgoing pitch bend messages will produce the correct pitch.

Channel: When **Voice Mode (page 15)** is set to 'Mono' this parameter sets the MIDI channel that outgoing MIDI messages will use. Drag all the way up for a "Keep Chan" option which will not adjust the MIDI channel of incoming messages. This may be useful for e.g. playing a sampled monophonic instrument with different articulations assigned to separate MIDI channels. When **Voice Mode (page 15)** is set to 'Poly', a consecutive range of MIDI channels can be set. Each note played will be assigned to a separate MIDI channel and the correct pitch bend messages to adjust each note will be sent on the corresponding channels.

Master Chan: This parameter is only available when **Voice Mode (page 15)** is set to 'MPE' and specifies the MPE master channel. MPE uses either channel 1 or 16 to transfer global MIDI data. The remaining channels are used for note and voice-specific MIDI data. Channel 1 is most commonly used as the master channel but the user guide for the synth being retuned should be checked to verify. Note that the MPE specification allows for channels 1 and 16 to be used simultaneously as master channels in a 'two-zone' configuration, but MTS-ESP MIDI Client does not support this at present.

MPE Receive: This parameter is only available when **Voice Mode (page 15)** is set to 'MPE'. It should ONLY be set to 'On' if MTS-ESP MIDI Client is receiving MPE MIDI data e.g. from an MPE MIDI controller. When on, MTS-ESP MIDI Client will not perform any MIDI channel reallocation, as notes will already be assigned to different channels.

PB Rate: This parameter limits the send rate of MIDI pitch bend messages generated by the MTS-ESP MIDI Client glide, or by real-time tuning changes in MTS-ESP Master. Higher values reduce MIDI traffic but will start to introduce audible pitch quantization. An initial setting of 10ms is recommended.

This method generates MIDI SysEx messages to transfer tuning data to a hardware synth. The MIDI Tuning Standard (MTS) is a specification agreed to by the MIDI Manufacturers Association that defines a few different formats of SysEx message for tuning data.

Using this method avoids the limitations of using MIDI pitch bend messages for retuning. The retuning range is very large, pitch bend and glide are unaffected and polyphony is supported.

Real-time tuning changes may however pose issues. MTS SysEx messages contain a flag which indicates whether they are 'real time' or 'non-real time'. If a synth receives and accepts a 'real time' format message it is reasonable to assume that it will immediately update the pitch of any held notes, although in practice this is not always the case.

If 'real time' messages are accepted, issues may still occur as a result of a large volume of SysEx messages being transmitted. Some DAWs, MIDI interfaces and MIDI hardware devices can struggle with too much SysEx traffic. To help avoid this, MTS-ESP MIDI Client offers a **SysEx Throttle (page 20)** parameter to limit the rate at which MTS SysEx messages are sent.

If a synth only accepts 'non-real time' format messages, it can be assumed that real time tuning changes are not possible. For this case, MTS-ESP MIDI Client offers a way to use MIDI pitch bend messages in combination with MTS SysEx messages to allow for real time tuning changes (see **Use MIDI PB (page 19)**).

About MTS Message Formats

Using MTS SysEx messages will require some understanding of how they are formatted. Full details are available from the MIDI Association website and an account will need to be created to access the relevant document: **https://www.midi.org/specifications-old/item/the-midi-1-0-specification**.

MTS SysEx messages all start with the same bits of information as follows (indicated here as hexadecimal bytes):

1. **F0h:** SysEx message start flag.
2. **7Eh or 7Fh:** This byte flags the message as 'real-time' (7Eh) or 'non-real-time' (7Fh). The expectation is that real-time messages will update tuning whilst a note is held.
3. **<device ID>:** This byte can be used to identify a specific device that the message should apply to. A value of 7Fh indicates that all devices should accept the message, however some devices may however only accept a value of 00h.
4. **08h:** This byte identifies the message as an MTS SysEx message.
5. **MTS ID:** This byte will be numbered between 00h and 09h. There are ten types of MTS SysEx message and this indicates which type it is. Two (00h and 03h) are requests that can be sent to a device to query its tuning. The rest are different ways of transferring tuning data to a device.

The user guide for a hardware synth will usually tell you if it supports MTS and may either give an example message or explicitly state the MTS format(s) it accepts. By examining the initial bytes of an example message and referencing the list above it should be possible to work out which format of MTS message it accepts. We have included presets for all formats which can be accessed from the **Preset Menu (page 12)** in MTS-ESP MIDI Client. In addition there are a number of presets for specific synthesisers for quicker set up.

Output Parameters

Output: This menu allows you to specify whether MTS SysEx messages are sent to the host or directly to a MIDI device or virtual MIDI port. This is useful in hosts which block SysEx data.

Note that on Windows it is unlikely that the same MIDI port can be enabled in both the host and the plug-in.

Auto Send:

- **Off:** MTS SysEx messages are only sent when the 'Send Dump' button is clicked.
- **Scale Select:** MTS SysEx messages are sent automatically only when a new scale is selected in the MTS-ESP Master plug-in.
- **On:** MTS SysEx messages are sent automatically upon any tuning change in the MTS-ESP Master plug-in.

Use MIDI PB: This is only available when **Auto Send (page 19)** is set to 'Off' or 'Scale Select'. It allows MIDI pitch bend messages to be used for real-time tuning changes in combination with MTS SysEx messages. It is useful where a synth does not accept real-time MTS messages, or where real-time MTS messages generate too much SysEx traffic for a device to handle.

To use this option, first switch the **Retune Method (page 14)** to 'MIDI PB' and set the parameters as required (see the 'Method: MIDI PB' section for details). Then switch **Retune Method (page 14)** back to 'MTS SysEx', set **Use MIDI PB** to 'On' and click **Send Dump (page 20)** to make sure the tuning table on the device matches that in MTS-ESP MIDI Client.

Now any real time tuning changes will be applied using MIDI pitch bend messages to adjust the note from its value in the stored tuning table. The limitations of using MIDI pitch bend outlined in the **Overview Of Use (page 10)** section still apply.

Note that whilst **Retune Method (page 14)** is set to 'MIDI PB', no MTS SysEx messages will be sent, even if **Use MIDI PB** is set to 'On'.

Throttle: This is only available when **Auto Send (page 19)** is set to 'On'. This limits the send rate of MTS SysEx messages generated as a result of real-time tuning changes in the MTS-ESP Master plug-in. An initial value of 10ms is recommended. Lower values will increase accuracy but introduce more SysEx traffic. If a system or device appears to struggle with the volume of SysEx data, increase this value. Higher values can result in audible pitch quantization in which case it may be preferable to use a combination of MTS SysEx and MIDI pitch bend messages (see **Use MIDI PB (page 19)** above).

Send Dump: Click to manually send a dump of the current tuning.

Device ID Parameters

MTS SysEx messages include a device ID, which is checked by a receiving device to see if it should process or ignore the message. A special value of 127 (or 7F in hex) is used to indicate the message applies to all devices, although some devices may require this be set to 0.

Ctrl/Cmd-Shift click on the device ID value to switch between decimal and hexadecimal display. A 'h' will be displayed after the value if hexadecimal display is enabled.

Message Parameters

Format: This parameter selects the MTS SysEx message format. Check the user guide for a hardware synth to see which format(s) it will accept before setting this parameter.

- **Bulk:** MTS message formats 1 and 4. This format contains frequency data for all 128 MIDI notes in a single message. It is probably the most commonly used format. It is not suited for real-time tuning changes.
- **Single:** MTS message formats 2, 7 and 8. This format contains the frequency data for a single MIDI note and, with some synths, can be used for real-time tuning changes. The name is a little misleading as it is possible to include frequency data for multiple notes in one 'single' note change message, up to a maximum of 127. To minimise SysEx traffic, MTS-MIDI Client only sends single note change messages to update the tuning, where required, of notes currently playing or those about to be triggered.
- **Bulk+Single:** This is not a message format but rather a mode that can be used with a device that accepts both Bulk and Single format messages. A Bulk message will be sent whenever a new scale is selected in the MTS-ESP Master plug-in but any real time tuning changes will generate Single note change messages.
- **Scale/Oct:** MTS message formats 5 and 6. This message format allows detuning of the standard 12-TET tunings by a maximum of -100/+100 cents. It may therefore be of limited use, but could be suited to exploring different 12-tone temperaments. It is not suited for real-time changes.
- **Scale/Oct Ext:** MTS message formats 8 and 9. A later revision of the MTS specification built upon the Scale/Oct format. It added a real time option and the ability to apply retuning to specific MIDI channels.

Bank: This parameter is NOT available when **Format (page 20)** is set to 'Scale/Oct Ext'. MTS messages can include a 'bank' number to allow devices to store banks of tuning tables. If the device you are retuning supports it, set this to the tuning table bank containing the tuning table you wish to overwrite, otherwise set it to 'Off'.

Note that the original MTS specification did not include a bank number, so some devices may not expect it in the message at all. Check the user guide for a synth to see if this is the case and, if so, set **Bank** to 'Off'.

A bank number must always be included when **Format (page 20)** is set to 'Scale/Oct', so the 'Off' option is not available with this format.

Prog: This parameter is NOT available when **Format (page 20)** is set to 'Scale/Oct Ext'. MTS messages can include a 'program' number to allow devices to store multiple tuning tables. If the device you are retuning supports it, set this to the tuning table program number you wish to overwrite, otherwise set it to 0.

Checksum: This parameter is only available when **Format (page 20)** is set to 'Bulk', 'Bulk+Single' or 'Scale/Oct' and applies only to Bulk and Scale/Oct format messages. If set to 'Calculate', the checksum byte at the end of the message will be calculated as specified in the MIDI Tuning Standard specification. If set to '7Fh', the checksum byte will be fixed at 7F (127), which is required by some synthesisers, notably units from E-mu.

Notes: This parameter is only available when **Format (page 20)** is set to 'Single' or 'Bulk+Single' and applies only to Single note change messages. It limits the number of frequency changes that a Single note change message can contain and may need to be adjusted when making real time tuning changes in the MTS-ESP Master plug-in.

Lower values result in shorter SysEx messages, but a higher volume of them. Larger values result in longer but fewer SysEx messages. The optimum value will depend on how a system or device handles SysEx messages and setting it may require some experimentation.
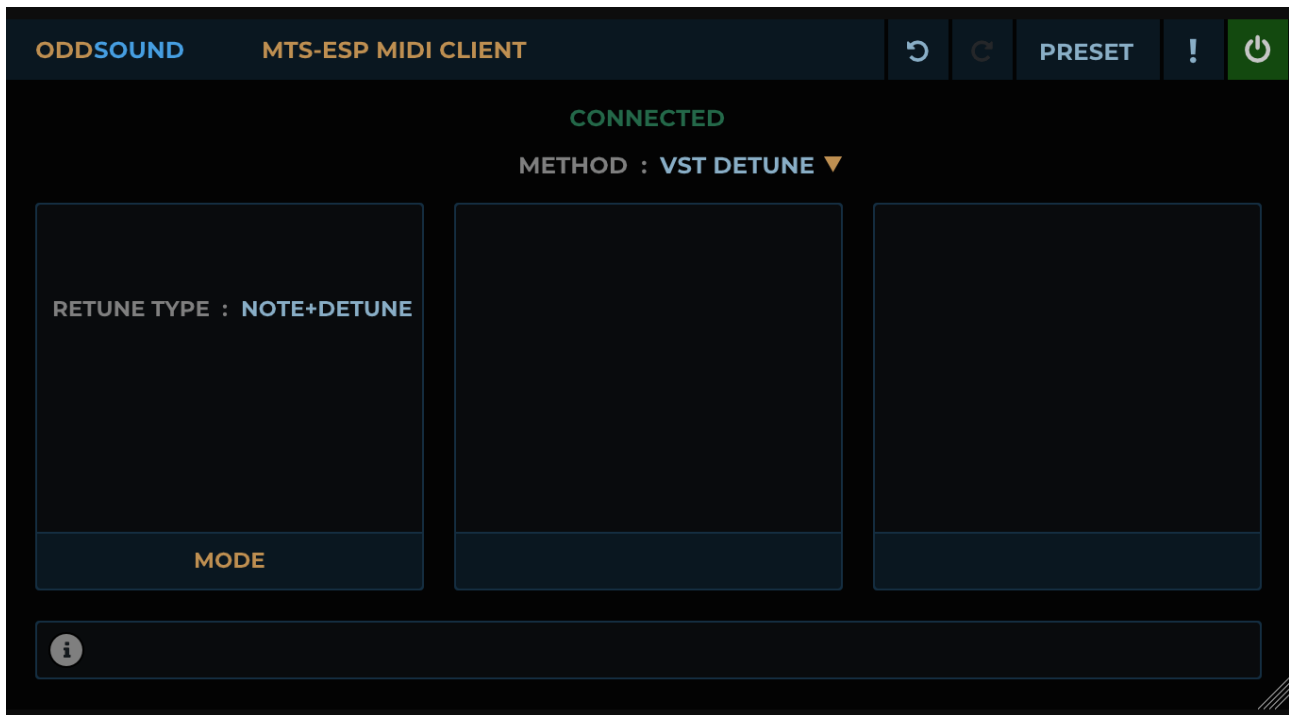
Realtime: This parameter is only available when **Format (page 20)** is set to 'Single', 'Bulk+Single' or 'Scale/Oct Ext'. It only applies to Single note change or Scale/Oct Ext format messages, where the MTS specification allows for the header flag to be set to either 'real time' (7Fh) or 'non-real time' (7Eh). The user guide for a synth should indicate which is accepted.

The original MTS specification did not allow a choice for Single note messages, just as it did not include a bank number (see above). Therefore if **Bank (page 20)** is set to 'Off', the **Realtime** parameter will be hidden and the header flag for single note messages always set to 'real time', in accordance with the original MTS specification.

Byte Format: This parameter is only available when **Format (page 20)** is set to 'Scale/Oct' or 'Scale/Oct Ext'. It specifies whether the frequency data is sent using 1 or 2 bytes. The user guide for a synth should indicate which is accepted.

The 1 byte format has a range of -64 to +63 cents and a maximum accuracy of 1 cent. The 2 byte format has a range of -100 to +100 cents and a maximum accuracy of ~0.01 cents.

Channel: This parameter is only available when **Format (page 20)** is set to 'Scale/Oct Ext'. Set the range of MIDI channels that the retuning will apply to.

ODDSOUND   **MTS-ESP MIDI CLIENT**   ↺ C **PRESET** ! ⏻

CONNECTED

METHOD : VST DETUNE ▼

RETUNE TYPE : NOTE+DETUNE

MODE

ⓘ

When MIDI messages are passed between a plug-in and its host, there is usually some extra data accompanying the MIDI message.

The VST2, VST3 and VST Module Architecture (or VST-MA, for MIDI FX in Cubase/Nuendo) specifications include a 'detune' value which allows per-note tuning adjustment. MTS-ESP MIDI Client can set this value accordingly to retune a VST2 or VST3 plug-in synth.

The range of this value varies between the specifications. For VST2 it is -64 to +63 cents and for VST-MA it is -100 to +100 cents. In both these specifications it is represented by an integer number, so retuning is accurate to 1 cent.

In the VST3 specification, detune in cents is stored as a 32-bit floating point number and there is no limit to the range, making it both more useful and more accurate.

This method has the advantages that it doesn't use MIDI pitch bend messages, so pitch bend can be used as usual, and it works for polyphonic parts. There are disadvantages, however they may not pose a problem depending on requirements.

The detune is applied at note-on events only, so no real-time changes to tuning will occur whilst a note is held.

Glide may not work correctly if the synth does not take the detune value into account when calculating glide.

Finally, the detune value is not guaranteed to be used by all plug-ins or hosts. A number ignore it altogether, others may ignore it for one plug-in format but not the other. Note it will not work at all for AU or AAX plug-ins. Unfortunately support for VST detune is not usually documented so some trial and error may be required in finding out if it will work.

There is only a single parameter available in this mode:

Retune Type: If set to 'Note+Detune', notes are transposed to that nearest the target pitch before the retuning amount is calculated. This should be used when retuning VST2 plug-ins, or when using the VST2 or VST MIDI FX versions of MTS-ESP MIDI Client, as the retuning range is very limited.

If using the VST3 version of MTS-ESP MIDI Client to retune a VST3 synth, this parameter should be set to 'Detune Only', as the retuning range is unlimited and no transposition is necessary to reach the target pitch. This is the ideal use case for this method.

Yamaha FB-01

This method is intended to be used with the Yamaha FB-01 FM synthesiser. It has a tuning implementation not found on any other Yamaha synths that lends itself quite well to dynamic tuning.

There are no parameters associated with this method. It works by replacing all note-on and note-off messages with unique SysEx format note-on and note-off messages that the FB-01 can understand. These SysEx messages include a 'fractional tuning' byte, specifying an offset in cents from the supplied note number. This allows on-the-fly retuning with 1 cent accuracy. Retuning is only applied at the start of a note and cannot be changed whilst a note is held.

This method should be used in a DAW that supports plug-ins sending SysEx data. No 'Output' option is provided to route the SysEx messages directly to a MIDI output port, as with the **MTS SysEx method (page 18)**, as the note events must be sent through the DAW to preserve timing.

Waldorf Wave/Microwave I

This method is intended to be used with the Waldorf Wave and Microwave I synthesisers (not the Microwave II, XT or XTk). These synths have four tuning table locations that can be written to using a proprietary SysEx message format. Any MIDI note can be retuned to the pitch of any other MIDI note, with a fractional cents offset. This method therefore allows retuning with 1 cent accuracy.

Using this method is not unlike using the MTS SysEx Bulk dump format and some options from the MTS SysEx method are also available here, mostly functioning in exactly the same way. These include:

- **Output (page 19)**
- **Auto Send (page 19)**
- **Throttle (page 20)**
- **Send Dump (page 20)**
- **Device ID (page 20)**
- **Prog (page 20):** With this method, the range of the **Prog** parameter is limited to 0-3, according to the number of writable tuning tables on the hardware.
- **Realtime (page 20):** This parameter is only available for the Microwave I model. If enabled, single-note change messages will be sent to update tuning before a note-on message, if required. It is best used with **Auto Send** set to 'Scale'.

An additional parameter is included to select the synth model being retuned.

Live does not allow a MIDI plug-in to be inserted before an instrument plug-in in the Device View, therefore MTS-ESP MIDI Client cannot be directly inserted before a synth to retune it. Live does however allow a Max for Live MIDI effect to be placed before an instrument plug-in in the Device View, therefore we have created a Max for Live device that 'wraps' MTS-ESP MIDI Client.

Installing

When running the MTS-ESP Suite installer an option to install the Max for Live devices is offered.

Windows: If this option is checked, the installer will prompt for a location to install the devices. This should be set to the following folder inside the Live User Library:

User Library\Presets\MIDI Effects\Max MIDI Effect

OSX: When running the installer, click the 'Customise' button to see the Max for Live device install option. The devices must be installed into the Live User Library, the default location for which is ~/Music/Ableton/User Library. If the User Library location is different from this, or the MTS-ESP MIDI Client Max for Live devices do not appear in Live after installing, manually copy the Max for Live devices, plus the accompanying "MTS-ESP MIDI Client.vst" file, to the Live User Library. Copy them from:

/Library/Application Support/ODDSound/MTS-ESP MIDI Client/Max for Live MIDI Effects

to the following folder inside the Live User Library:

User Library/Presets/MIDI Effects/Max MIDI Effect

Using The Max for Live Devices For Pitch Bend Retuning

After installing, you should find two MTS-ESP MIDI Client Max for Live devices available in Live. One is an MPE version which allows polyphonic retuning, however this will only work in Live 11. The regular version can only be used for monophonic retuning since, when not using MPE, Live automatically sets MIDI messages to channel 1.

The Max for Live devices open in a 'reduced' view with only 4 controls. Click 'Show Params' for an expanded view showing all MTS-ESP MIDI Client parameters or, if you prefer to edit them from the plug-in window itself, click 'Open MTS-ESP MIDI Client'.

Monophonic Pitch Bend Retuning With A Virtual Instrument

To retune a virtual instrument, drag the MTS-ESP MIDI Client Max for Live device before it in the Device View. The default settings are set up for monophonic retuning and will work instantly if the pitch bend range of the instrument plug-in is set to +/-12 semitones.
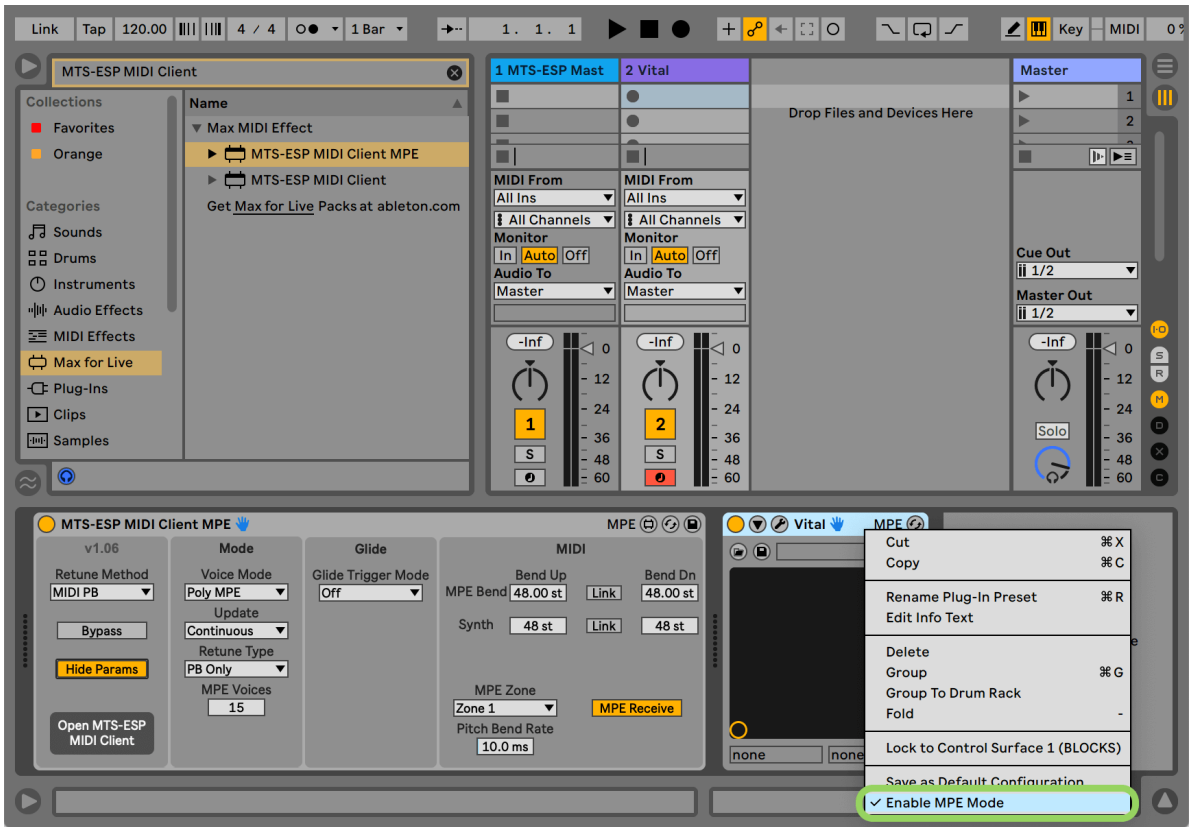
Polyphonic Pitch Bend Retuning With A Virtual Instrument That Supports MPE In Live 11

To polyphonically retune a virtual instrument which supports MPE or voice-per-channel, drag the MTS-ESP MIDI Client MPE Max for Live device before it in the Device View. The default settings will work instantly if the MPE pitch bend range of the virtual instrument is set to +/-48 semitones (this is usually the default).


Polyphonic retuning will work with Live's Wavetable, Simpler and Sampler instruments. For third-party plug-ins, ensure that MPE mode is enabled in Live by right-clicking on the device title bar and selecting 'Enable MPE Mode'. It may also be necessary to enable MPE in the plug-in itself, consult the plug-in's documentation for further information.

Polyphonic Pitch Bend Retuning With A Virtual Instrument In Live 10 And Below

To polyphonically retune a third-party virtual instrument which supports MPE or voice-per-channel, it must be loaded along with MTS-ESP MIDI Client into a hoster plug-in i.e. one that is itself capable of loading plug-ins. See the end of this manual from some recommended **hoster plug-ins (page 54)**. This gets round the issue of Live setting all MIDI data to channel 1.
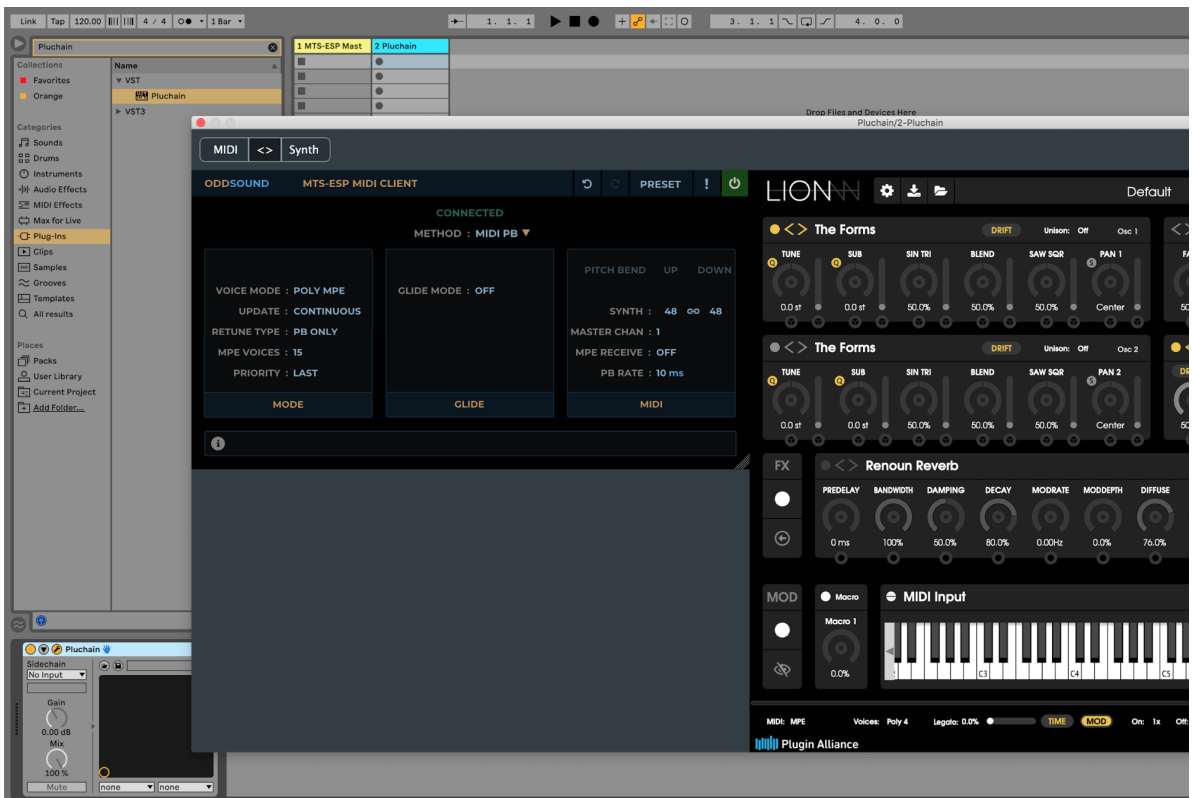
Using MTS SysEx Mode In Live

Live doesn't support SysEx so to use MTS-ESP MIDI Client in MTS SysEx mode you will need to route SysEx data directly to a hardware device using the **Output (page 19)** port option in MTS-ESP MIDI Client. Note this option is not available from the Max for Live device panel and it's state will not be saved when using the MTS-ESP MIDI Client Max for Live devices, so it is recommended to use the VST2 version of the plug-in.

Max For Live Device 'Couldn't Find Plug-In' Error

The following message will appear on the MTS-ESP MIDI Client Max for Live device panel if it can't find the MTS-ESP MIDI Client plug-in on disk:
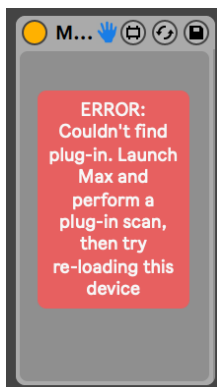
If this occurs, first make sure the VST2 version of the plug-in is installed. The easiest way to do this is to run the latest installer again, making sure that the VST2 format is selected. If the error message still appears, perform a plug-in scan in Max as follows:

- Launch Max. This can be done by clicking the Max Edit button in the device title bar of any Max for Live device.
- If on Windows, select Options->File Preferences and add your VST2 plug-ins folder to the paths list.
- Select Options->Preferences and perform a complete/slow plug-in scan, then restart Max and Live.
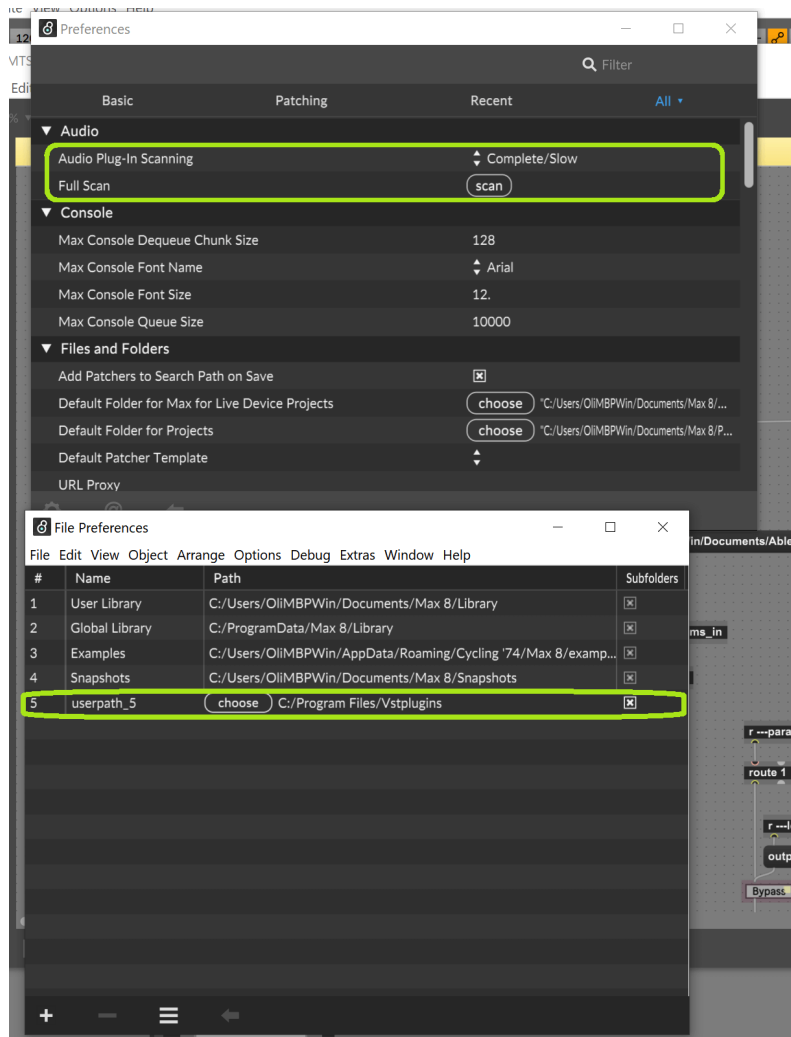
It is recommended to always use the VST2 version of MTS-ESP MIDI Client in Bitwig for optimal MIDI compatibility. To insert MTS-ESP MIDI Client before a virtual instrument in Bitwig it must be placed inside a Note FX Layer device.
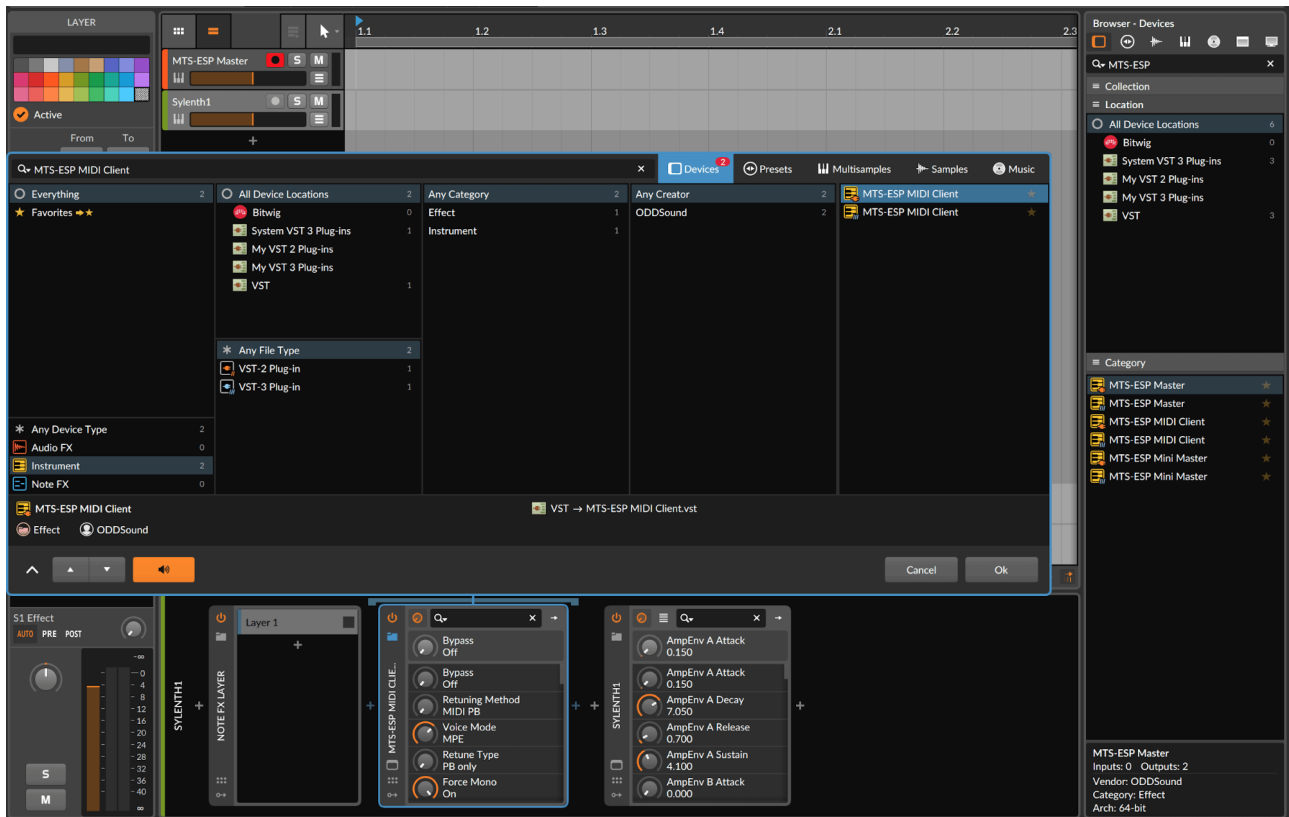
Polyphonic Pitch Bend Retuning With MPE And Bitwig Instruments

All Bitwig instruments support MPE and can be polyphonically retuned using MTS-ESP MIDI Client. To set this up:

- Add a Bitwig instrument to a track and insert MTS-ESP MIDI Client in a Note FX Layer before it, as shown above. In MTS-ESP MIDI Client select the 'MIDI Poly MPE Receive' preset, from the **preset menu (page 12)**.
- Enable MPE mode in Bitwig for MTS-ESP MIDI Client and ensure PB Range is set to +/-48.
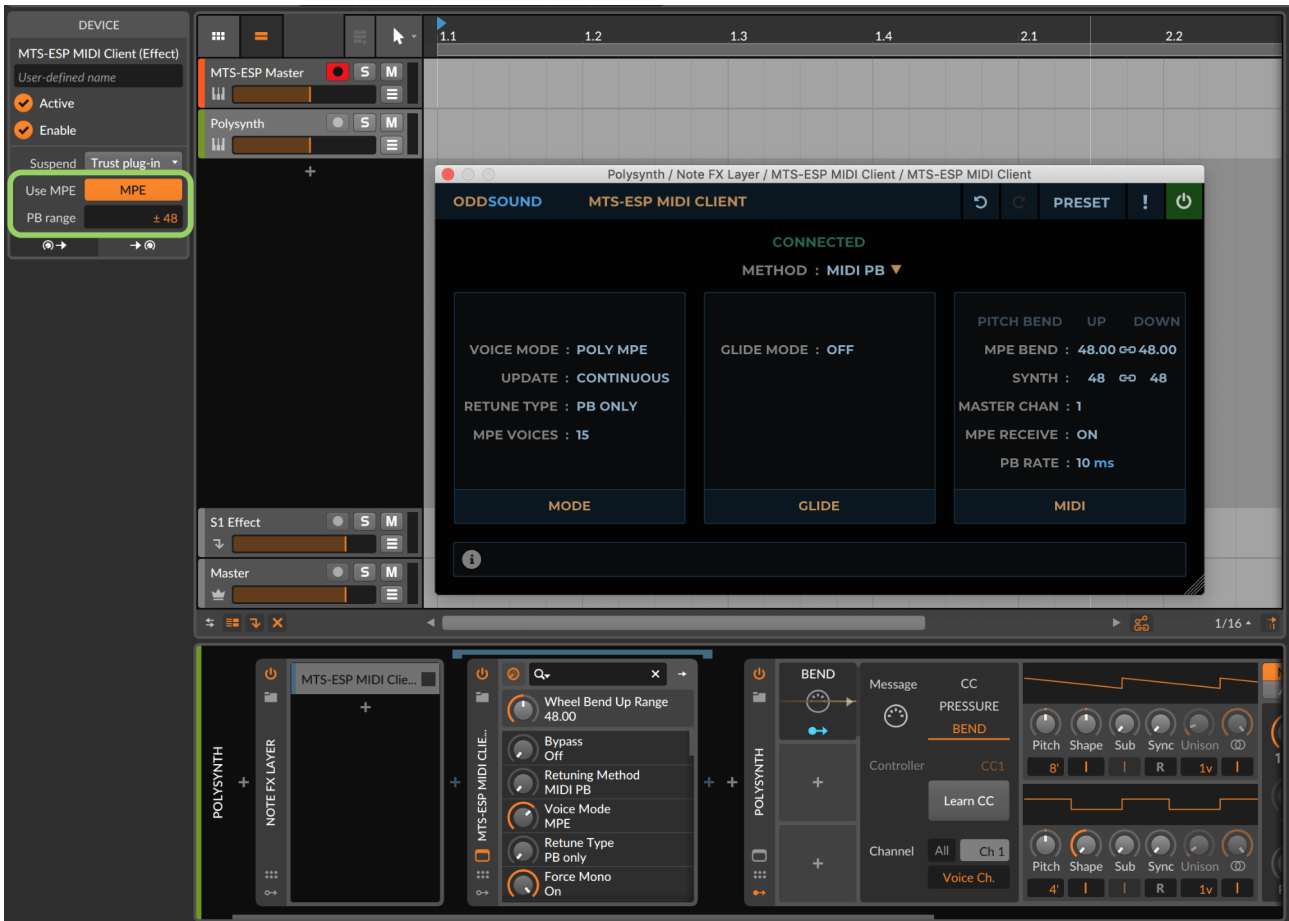
- Add a MIDI modulator to the Bitwig instrument, set message to BEND and, in the inspector, enable Per-Voice.

- In the MIDI modulator settings set Channel to 'Voice Ch.' (disable 'All' to be able to select this).

- Finally, assign the MIDI modulator to the instrument's Pitch parameter, with a range of +48 semitones.

Polyphonic Pitch Bend Retuning With MPE And Third Party Virtual Instruments

If using MPE to polyphonically retune a third party instrument plug-in, insert MTS-ESP MIDI Client in a Note FX Layer before the plug-in. Then, importantly, ensure that MPE is *not* enabled for both the instrument plug-in and MTS-ESP MIDI Client in Bitwig.
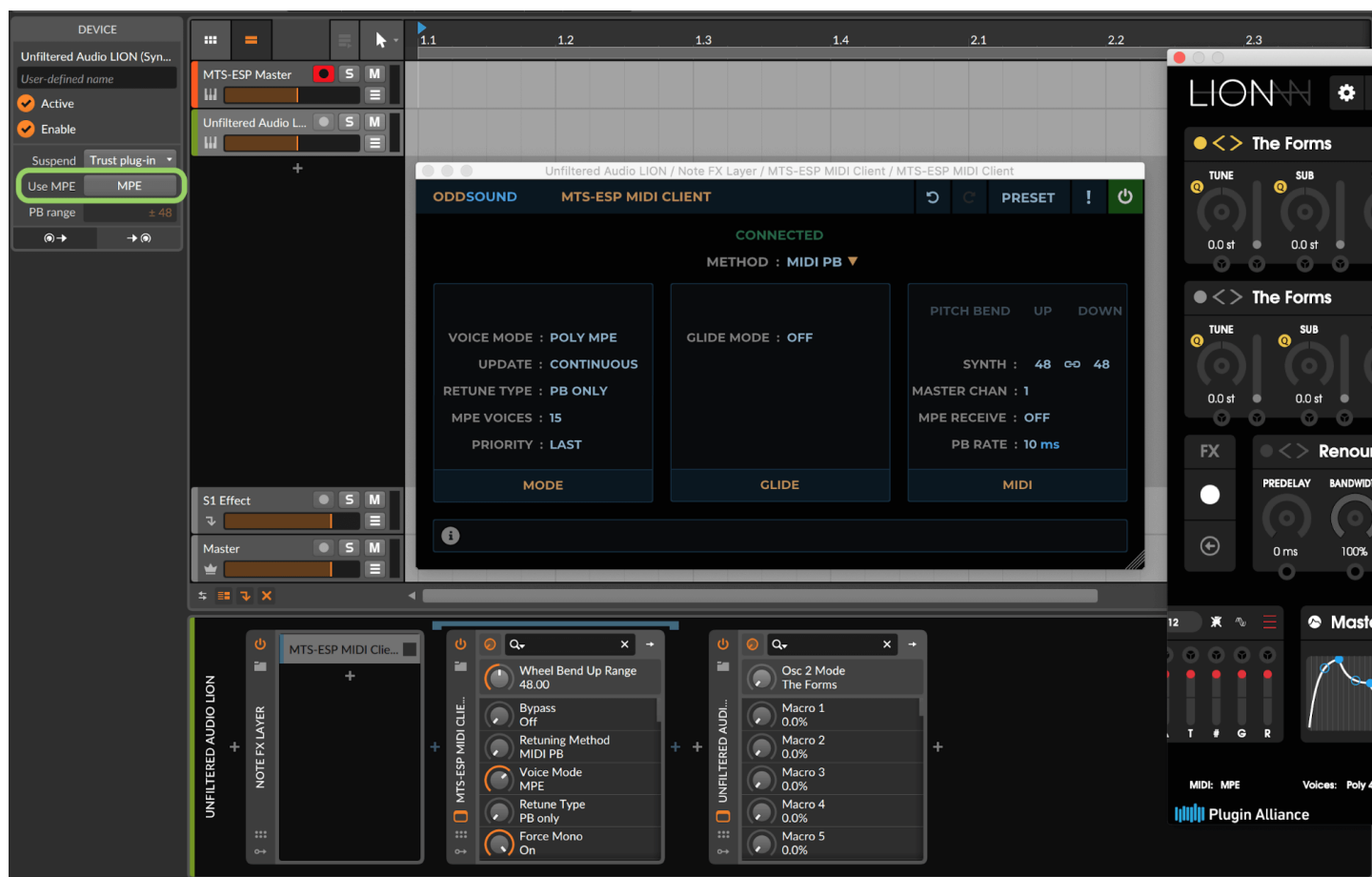
Using an MPE Controller With Polyphonic Pitch Bend Retuning

If using an MPE controller, MPE must be enabled for MTS-ESP MIDI Client in Bitwig, regardless of whether retuning a Bitwig instrument or a third party instrument plug-in. Also, crucially, **MPE Receive (page 17)**, must be set to 'On' in MTS-ESP MIDI Client.

Using MTS SysEx Mode In Bitwig

Bitwig blocks all SysEx data so to use MTS-ESP MIDI Client in MTS SysEx mode you will need to route MIDI data directly to a hardware device using the **Output (page 19)** port option in MTS-ESP MIDI Client.

It is recommended to always use the VST2 version of MTS-ESP MIDI Client in Cakewalk for optimal MIDI compatibility. To make sure it is available and not replaced with the VST3 version it may be necessary to disable the VST3 Migration options in the Cakewalk's VST preferences.
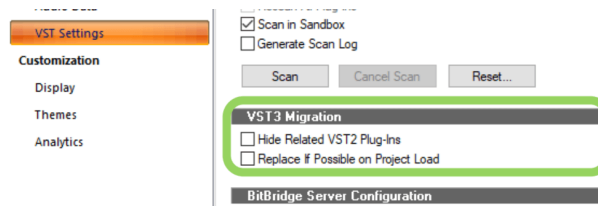
- Create an instrument track and add an instance of MTS-ESP MIDI Client to it. This is the track that will receive MIDI data and have notes programmed into it.
- Ensure MIDI output is enabled for MTS-ESP MIDI Client.
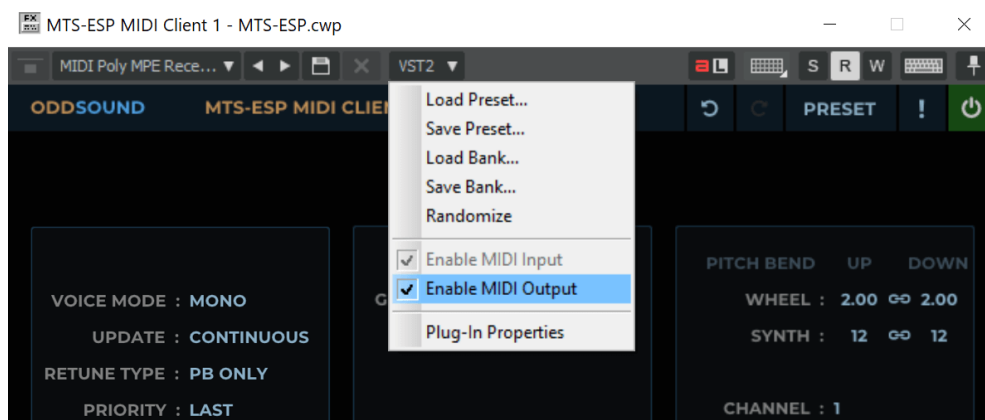
- To retune a virtual instrument, create another instrument track for it and set its MIDI input to the output of the MTS-ESP MIDI Client track with the 'MIDI Omni' option. Make sure input echo is enabled for the instrument track.
- To retune a hardware instrument create a MIDI track, similarly setting its MIDI input to the output of the MTS-ESP MIDI Client track and enabling input echo. Set it's MIDI output to the desired device or port.
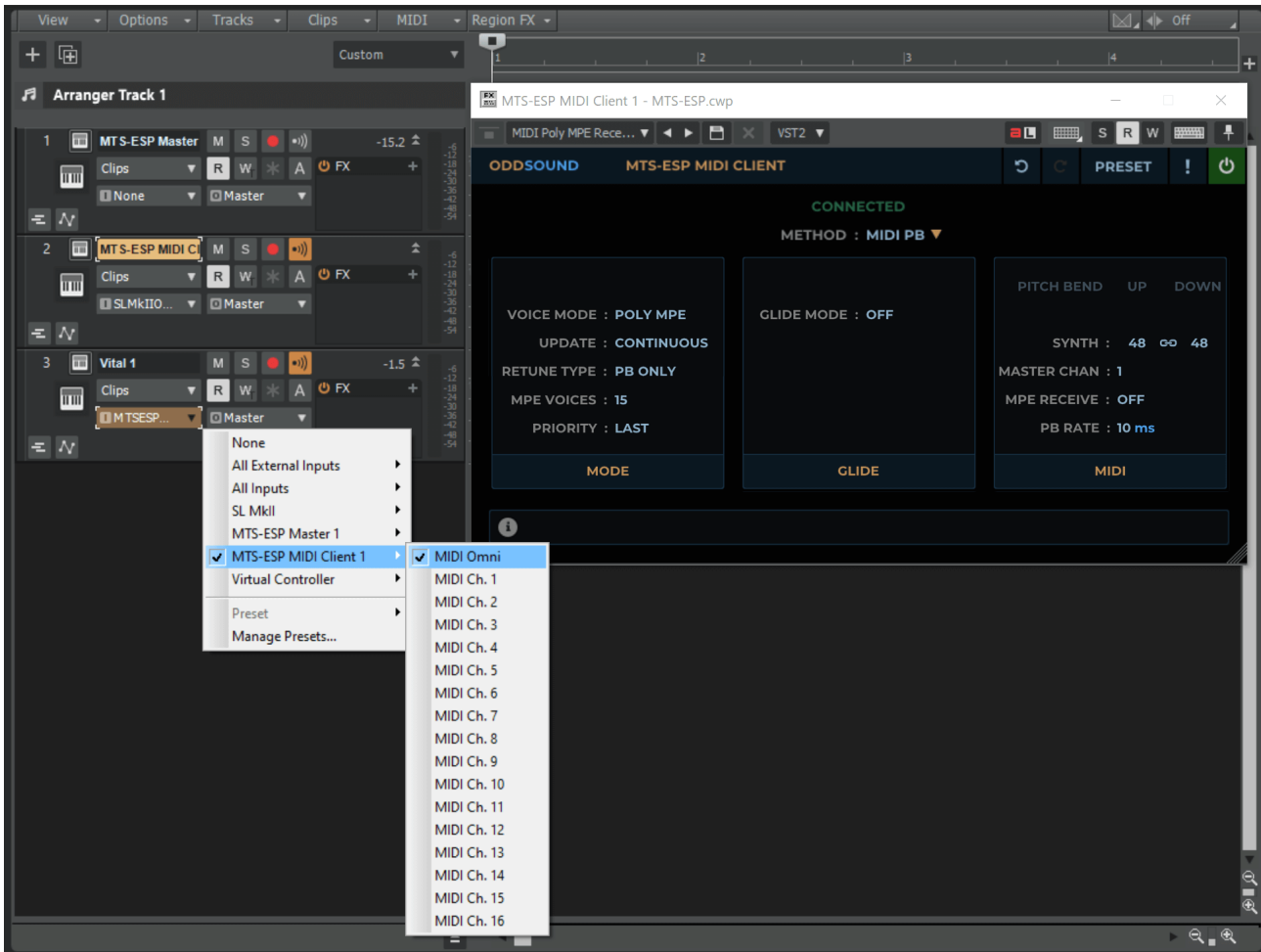
Using MTS SysEx Mode In Cakewalk

Cakewalk doesn't support SysEx data being sent from a VST plug-in so use the **Output (page 19)** port option in MTS-ESP MIDI Client to send SysEx messages directly to a physical MIDI output port.

About The VST-MA MIDI Effect Version

We have created a version of MTS-ESP MIDI Client specifically for use with Cubase/Nuendo using Steinberg's VST Module Architecture (VST-MA) SDK. This allows MTS-ESP MIDI Client to be inserted as a MIDI insert on a MIDI or instrument track.

The VST-MA SDK seems to have received little support in recent times and consequently there are some known issues with this version of MTS-ESP MIDI Client, therefore we provide it as an 'experimental feature'. If using MTS SysEx mode everything should work fine, however it may be preferable to use the regular VST2 or VST3 version when using the MIDI PB or VST Detune methods.

Installing

When running the MTS-ESP Suite installer an option to install the VST-MA MIDI effect version is offered. On OSX, click the 'Customise' button to see the VST-MA MIDI effect install option.

Retuning Using The VST or VST3 Version Of MTS-ESP MIDI Client

- Create an instrument track and insert an instance of MTS-ESP MIDI Client. Ensure that MIDI channel is set to 'Any' in the Cubase/Nuendo inspector.
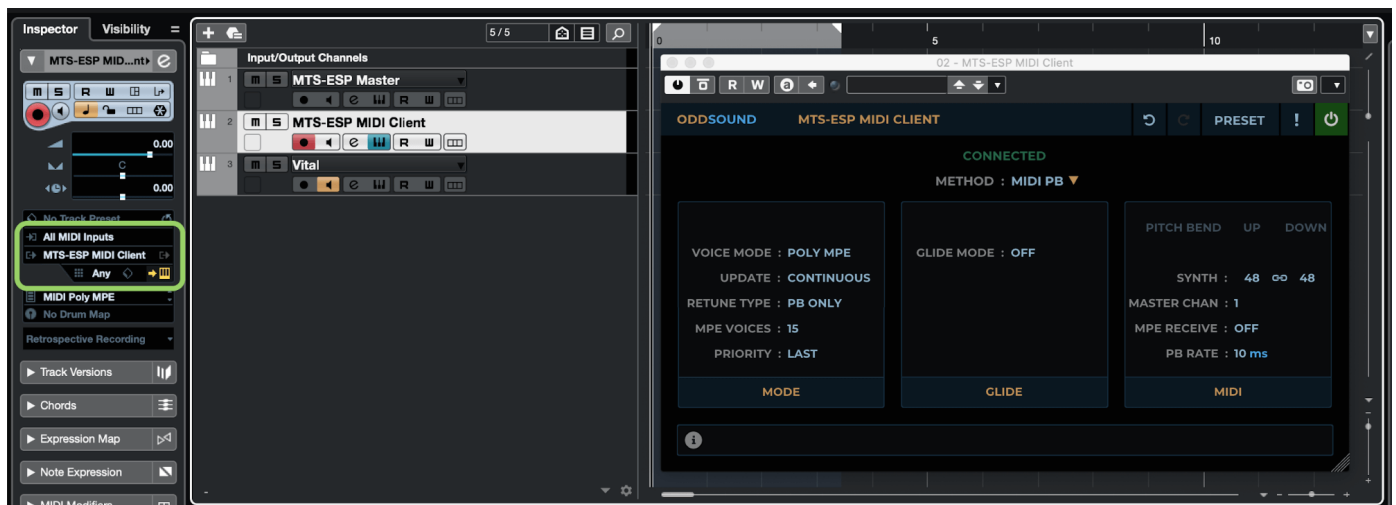

- Create another instrument track and insert an instance of the plug-in instrument you wish to retune, again making sure that MIDI channel is set to 'Any' in the Cubase/Nuendo inspector.
- Set the MIDI input for the plug-in instrument track to receive data from MTS-ESP MIDI Client. Make sure 'monitor', but not 'record', is enabled for the plug-in instrument track. Record-enable the MTS-ESP MIDI Client track to play notes in and hear the instrument retune.
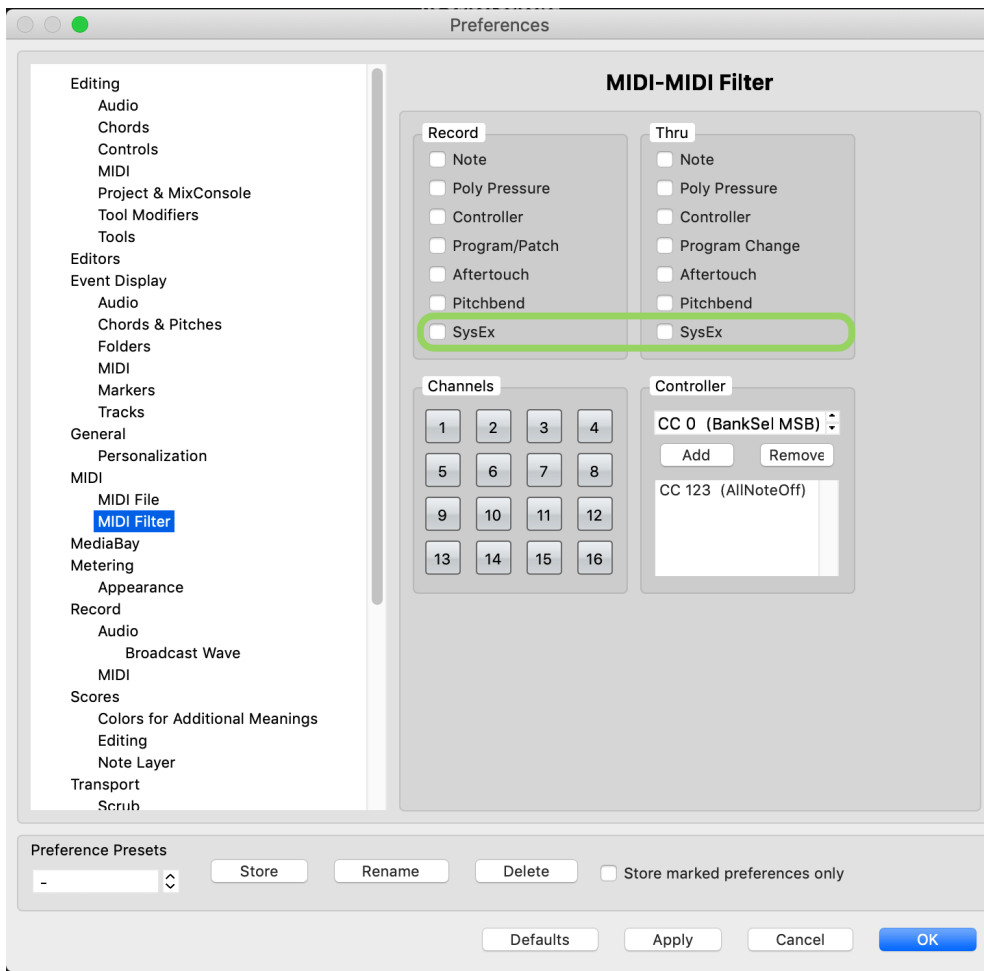
Retuning Using The VST-MA MIDI Effect Version Of MTS-ESP MIDI Client

Create an instrument or MIDI track and insert an instance of MTS-ESP MIDI Client in the 'MIDI Inserts' tab in the Cubase/Nuendo inspector.
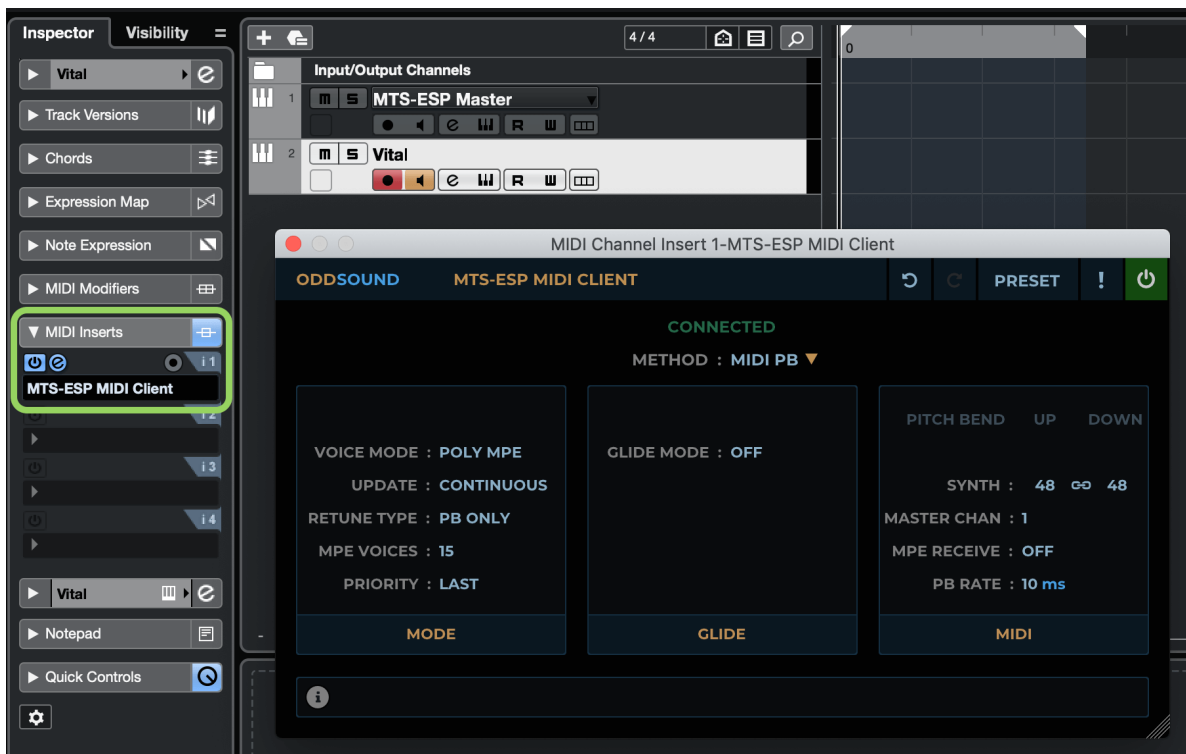

Using MTS SysEx Mode In Cubase/Nuendo

To use MTS SysEx mode, first make sure Cubase/Nuendo is not set to filter out SysEx messages. Go to Preferences->MIDI->MIDI Filter and make sure SysEx is unchecked, then click 'Apply' and 'OK'. With that set, create a MIDI track and insert an instance of MTS-ESP MIDI Client as a MIDI insert (see previous section).

Preferences

MIDI-MIDI Filter

Editing
    Audio
    Chords
    Controls
    MIDI
    Project & MixConsole
    Tool Modifiers
    Tools
Editors
Event Display
    Audio
    Chords & Pitches
    Folders
    MIDI
    Markers
    Tracks
General
    Personalization
MIDI
    MIDI File
    MIDI Filter
MediaBay
Metering
    Appearance
Record
    Audio
        Broadcast Wave
    MIDI
Scores
    Colors for Additional Meanings
    Editing
    Note Layer
Transport
    Scrub

Record
- [ ] Note
- [ ] Poly Pressure
- [ ] Controller
- [ ] Program/Patch
- [ ] Aftertouch
- [ ] Pitchbend
- [ ] SysEx

Thru
- [ ] Note
- [ ] Poly Pressure
- [ ] Controller
- [ ] Program Change
- [ ] Aftertouch
- [ ] Pitchbend
- [ ] SysEx

Channels

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Controller

CC 0 (BankSel MSB)

Add   Remove

CC 123 (AllNoteOff)

Preference Presets

-

Store   Rename   Delete   [ ] Store marked preferences only

Defaults   Apply   Cancel   OK

To be able to route MIDI data out of a plug-in in Digital Performer, you need to use the VST2 version. Go to Preferences->General->Audio plug-ins and create a new plug-in Set. Search for MTS-ESP MIDI Client, disable the Audio Unit and VST3 versions and enable the VST2 version, then click 'Done' and restart the audio engine.
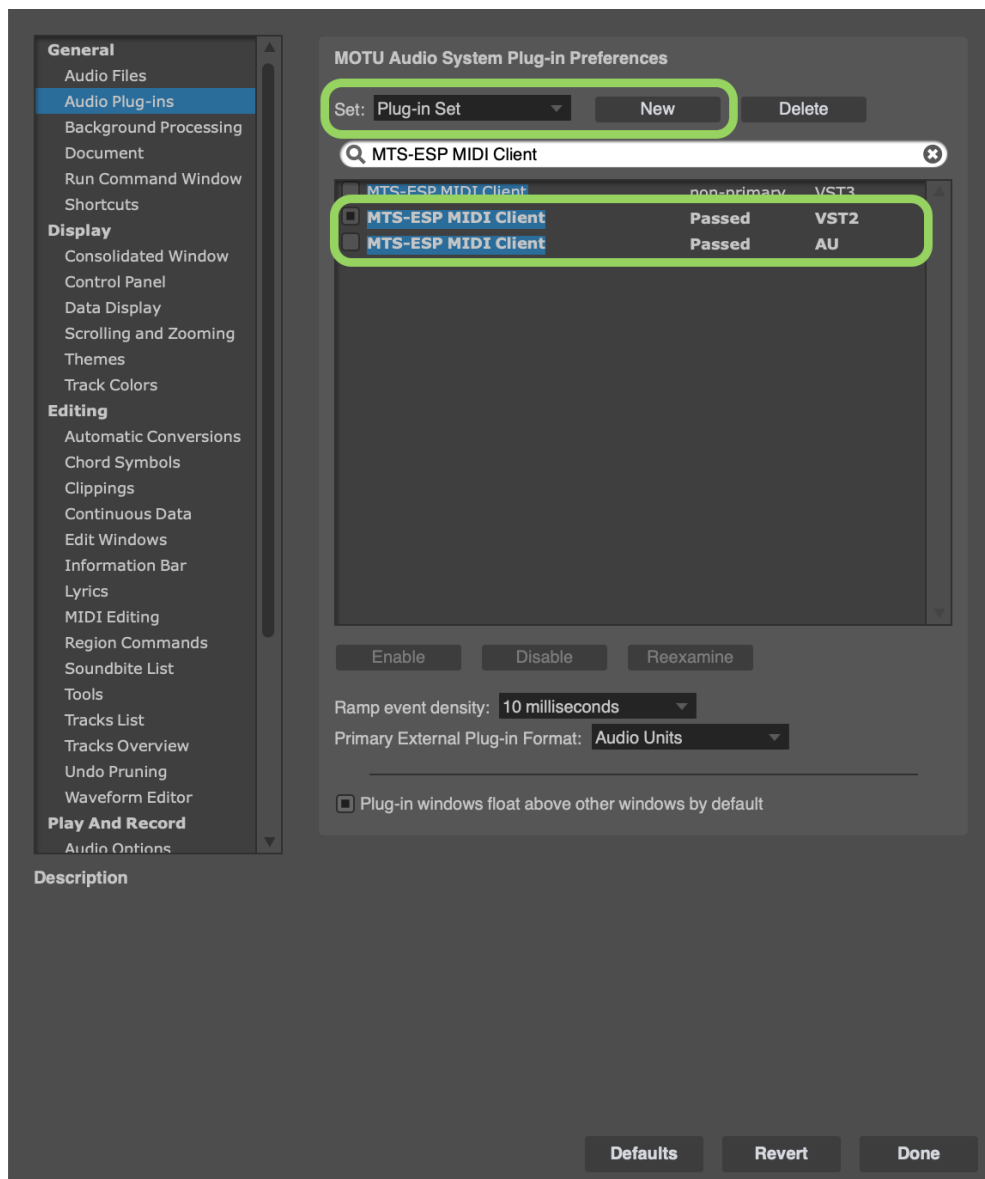
Monophonic Pitch Bend Retuning With A Virtual Instrument

Monophonic pitch bend retuning can be set up in Digital Performer as follows:

- Right-click in the track area and select 'Add Track->Instrument With Options...' (or press Option-Cmd-Shift-I).
- Add an instance of MTS-ESP MIDI Client and select to add a MIDI track for it. Set the input for the MIDI track as desired and make sure the output is set to MTS-ESP MIDI Client channel 1.
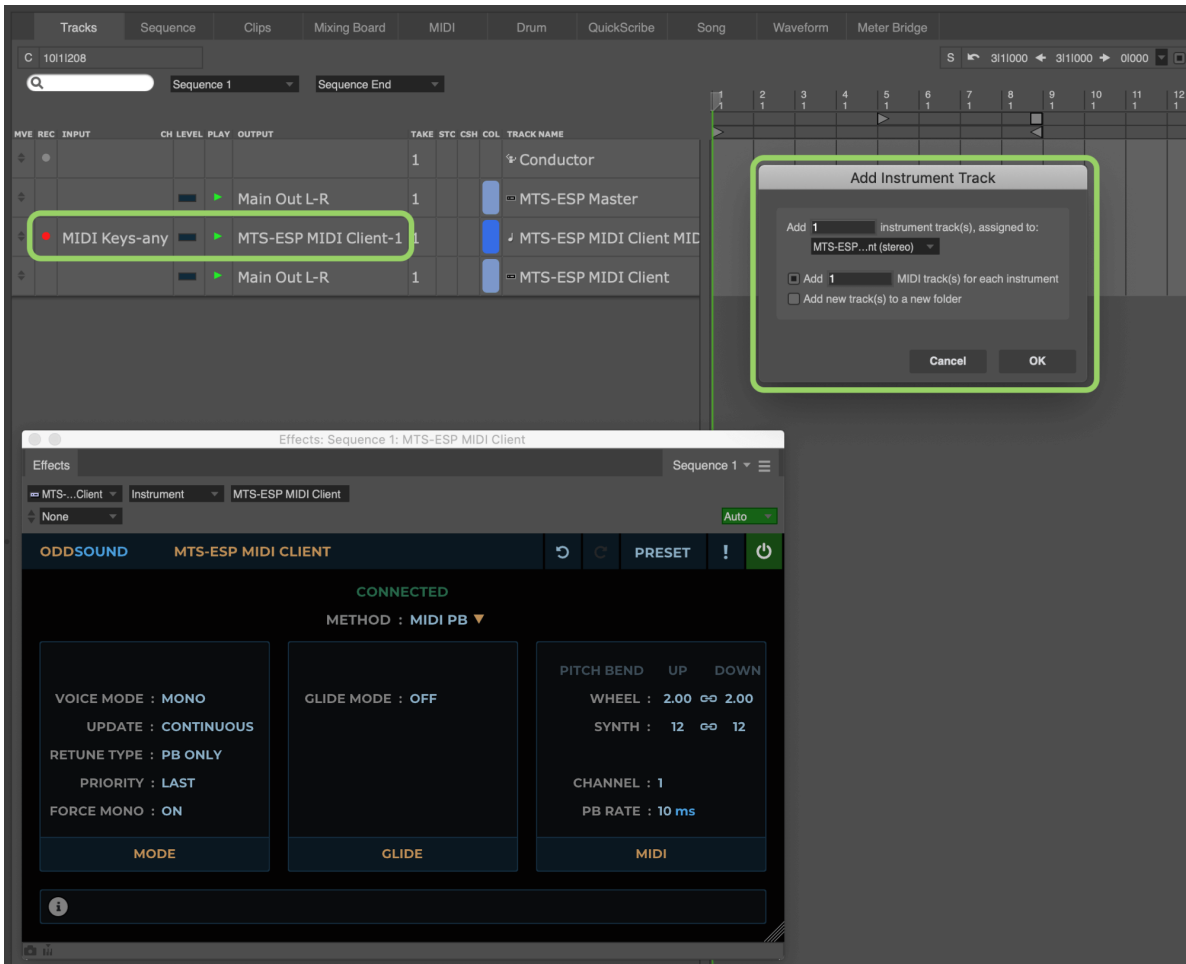
- Once again select 'Add Track->Instrument With Options…' and this time add an instance of the instrument plug-in you wish to retune, again creating a MIDI track for it.
- Ensure the input for the MIDI track is set to 'any' channel from MTS-ESP MIDI Client.
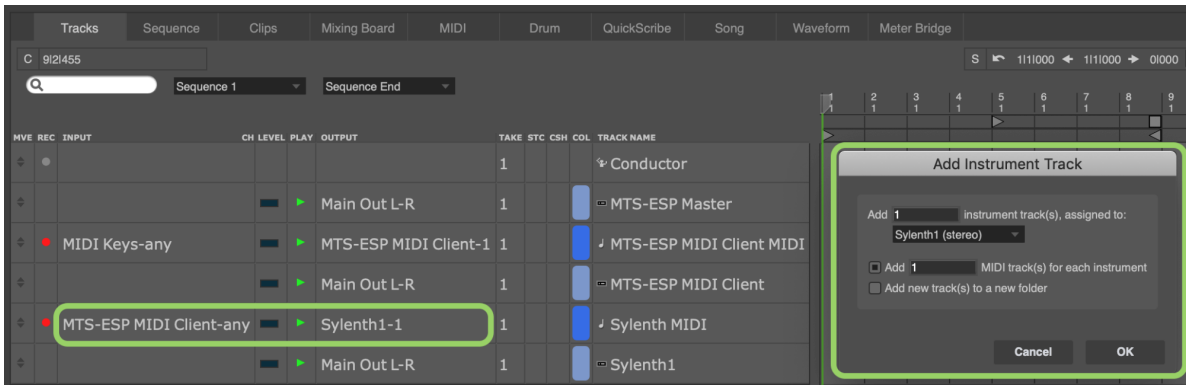
Unfortunately Digital Performer only allows single-channel routing of MIDI data to the input of a virtual instrument, so polyphonic retuning is not possible with this method.

Polyphonic Pitch Bend Retuning With A Virtual Instrument

For polyphonic retuning, or even monophonic retuning that requires less routing set up, we recommend using a hoster plug-in which allows a MIDI plug-in and a virtual instrument to be loaded together inside a single plug-in. See the end of this manual from some recommended **hoster plug-ins (page 54)**. This requires just a single instrument track and a single MIDI track to set up.
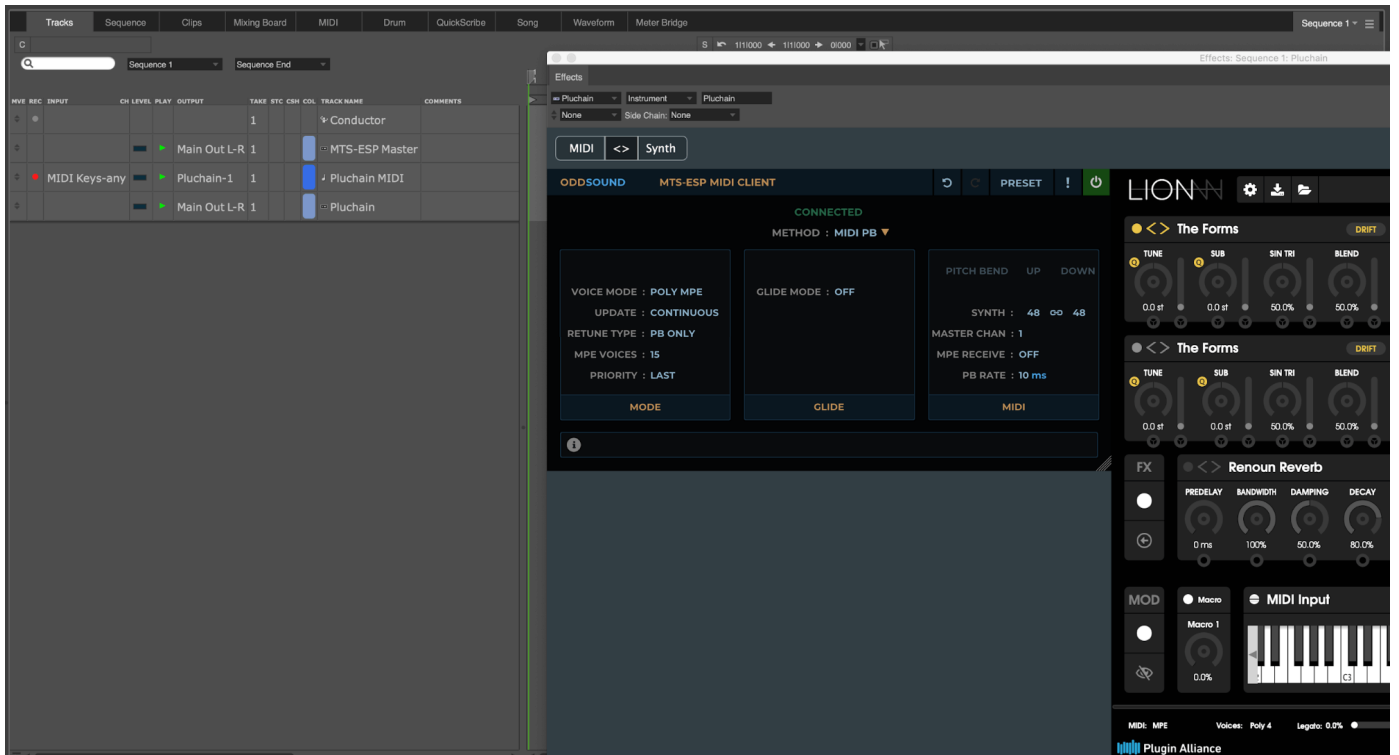
Using MTS SysEx Mode In Digital Performer

Digital Performer doesn't appear to support routing SysEx data from a VST plug-in to a MIDI output via a MIDI track, so use the **Output (page 19)** port option in MTS-ESP MIDI Client. In this case MIDI data doesn't actually need to be routed either into or out of MTS-ESP MIDI Client. Use a MIDI track to route data directly from a MIDI controller to a hardware MIDI out port.

Retuning Third Party Virtual Instruments

To retune a third-party virtual instrument in FL Studio, add an instance of MTS-ESP MIDI Client to the Channel Rack, along with an instance of the virtual instrument to be retuned. Open the windows for both and go to the plug-in settings. Set the MIDI Output Port number for MTS-ESP MIDI Client and the MIDI Input Port number for the virtual instrument to the same value. Make sure this port number is not being used by any other plug-ins in the session. Route note data to MTS-ESP MIDI Client to hear the synth retuned.
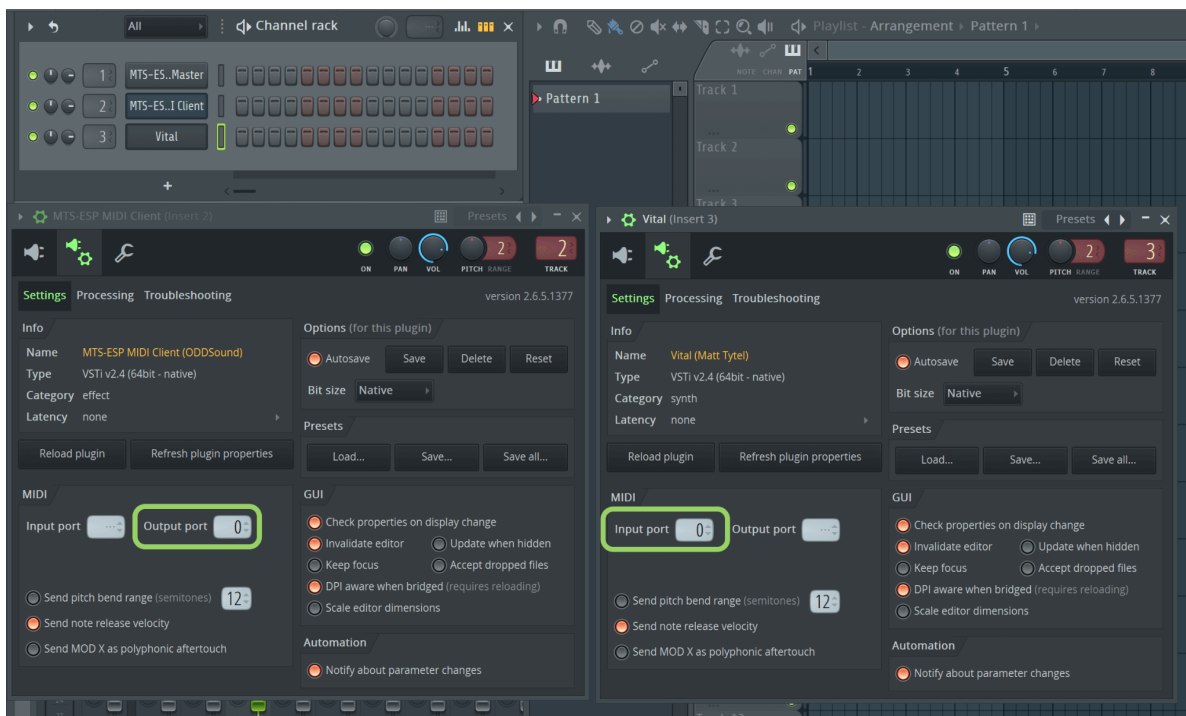
To retune a hardware instrument using MIDI pitch bend, set a number for the physical output port in FL Studio MIDI settings and then set MTS-ESP MIDI Client's MIDI Output port to match.

Using MTS SysEx Mode In FL Studio

FL Studio doesn't support SysEx data being sent from a VST plug-in so use the **Output (page 19)** port option in MTS-ESP MIDI Client to send SysEx messages directly to a physical MIDI output port.

Ordinarily, Kontakt is limited to monophonic pitch bend retuning as it doesn't support MPE or polyphonic pitch bend. There are however ways to get polyphonic pitch bend working.

Some sample libraries, notably those from Slate + Ash (**slateandash.com/**), have been designed to support MPE and will work with **Voice Mode (page 15)** set to 'Poly MPE'. With these, make sure **MPE Voices (page 16)** is set to match the number of MPE channels used by the library presets.

Polyphonic Pitch Bend Retuning Using Multiple Instruments

It is possible to set up polyphonic pitch bend retuning with multiple copies of the same instrument in a multi, each assigned to a different MIDI channel:

- Load an instrument into a new Kontakt instance/multi and set up as desired. Ensure pitch bend is assigned to pitch with a range of 12 semitones.

- Save the instrument, then load copies of the saved instrument into the multi, up to a total of 16, assigning each to a different MIDI channel. To minimise use of computer resources the number of copies should be limited to the number of voices required.

- Load an instance of MTS-ESP MIDI Client to process MIDI data going into Kontakt according to the relevant DAW setup instructions in this manual. If the DAW does not allow multi-channel MIDI routing (e.g. Ableton Live, Renoise, Studio One), it will be necessary to load MTS-ESP MIDI Client and Kontakt together inside a hoster plug-in. See the end of this manual from some recommended **hoster plug-ins (page 54)**.

- Select the 'MIDI Poly' preset in MTS-ESP MIDI Client. Set the **Channel (page 17)** range to match those set for the instruments in Kontakt and make sure the **Synth Pitch Bend (page 17)** range matches that in Kontakt, from the first step above.

Polyphonic Pitch Bend Retuning With Locked Libraries

Some Kontakt libraries are locked and will not allow pitch bend to be assigned to pitch as instructed in the first step in the previous section. In this case, follow these steps:

- Load new instance of Kontakt or create a new multi. Click the multi KSP button and in the first tab select Preset->Factory->Transform->Transformer.

- In the Transformer settings, set Pitch Bend as the 'transform' MIDI message and Controller as the 'into' MIDI message with any CC number of choice. Make sure MIDI channel options are set to 'As Played'.

- Load an instrument and set up as desired. Right click the Tune parameter and learn it to the same CC from the previous step (moving the pitch bend wheel on a MIDI controller should do this).

- In the MIDI Automation tab in the Kontakt browser find the assignment from the previous step and set 'From' to 47.2% and 'To' to 52.8%, meaning the CC will control pitch over a +/-2 semitone range.

- Save the instrument, then load copies of the saved instrument into the multi, up to a total of 16, assigning each to a different MIDI channel. To minimise use of computer resources the number of copies should be limited to the number of voices required.

- Load an instance of MTS-ESP MIDI Client to process MIDI data going into Kontakt according to the relevant DAW setup instructions in this manual. If the DAW does not allow multi-channel MIDI routing (e.g. Ableton Live, Renoise, Studio One), it will be necessary to load MTS-ESP MIDI Client and Kontakt together inside a hoster plug-in. See the end of this manual from some recommended .

- Select the 'MIDI Poly' preset in MTS-ESP MIDI Client. Set the **Channel (page 17)** range to match those set for the instruments in Kontakt and the **Synth Pitch Bend (page 17)** range is set to +/- 2 semitones.

- Finally in MTS-ESP MIDI Client set **Retune Type (page 15)** to 'Note+PB', so that notes are transposed to those nearest the target pitches before pitch bend retuning is applied. This setting is advised because of the small pitch bend range used here, necessary to achieve accurate retuning with only 128 CC steps. Note however that automating sliding pitch changes over large intervals in MTS-ESP Master will not work well with this setting enabled.
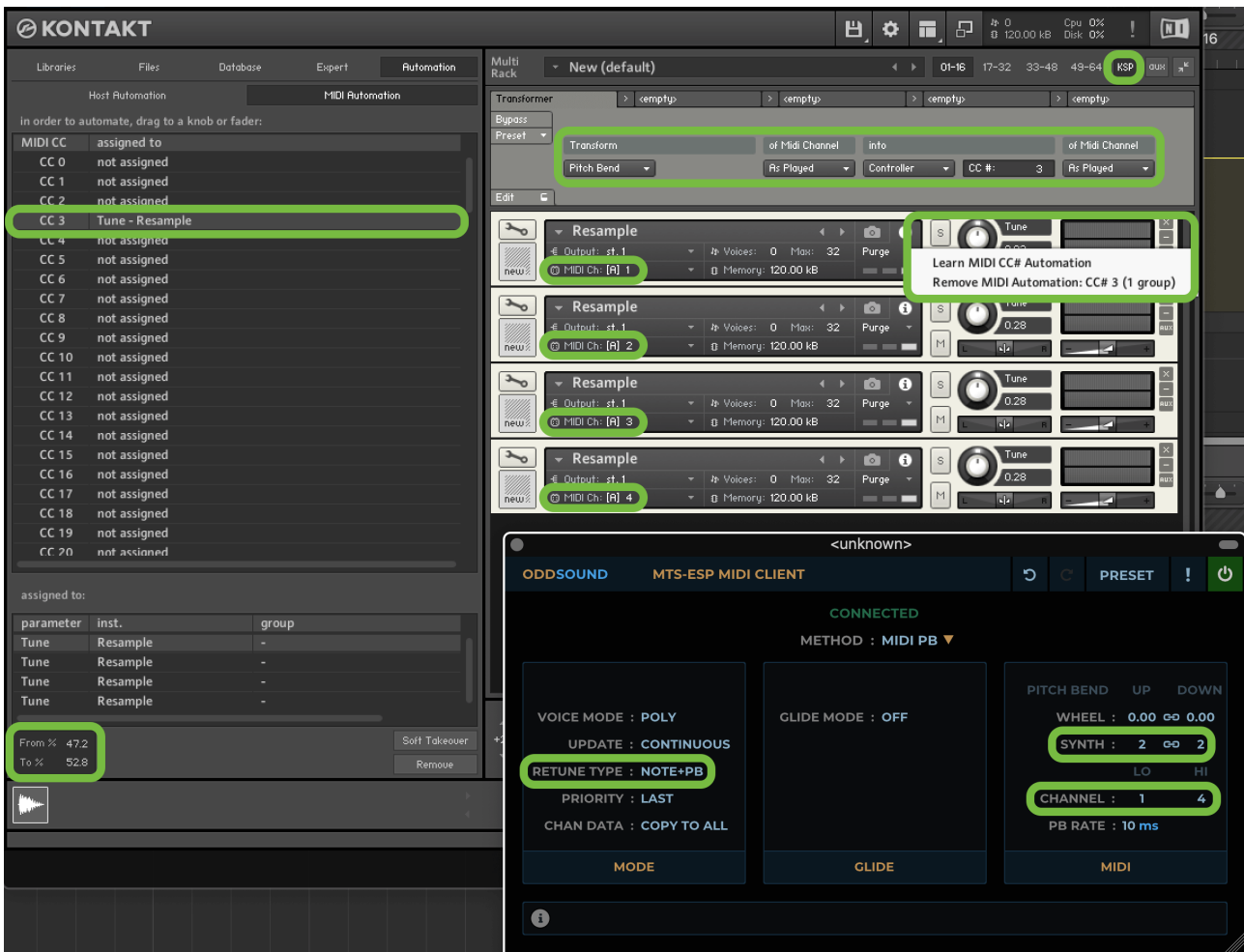
Logic Pro provides dedicated insert slots for MIDI plug-ins. Add an instance of MTS-ESP MIDI Client and set up as required.

When using the MTS-ESP Suite plug-ins in Logic, it is recommended to set a project up so that all plug-ins are loaded when it is opened. In Logic select File->Project Settings->General and make sure 'Only load plug-ins needed for project playback' is unchecked. This will ensure the MTS-ESP Suite plug-ins, along with any other plug-ins that support MTS-ESP, are loaded, connected and active when the project loads.

Retuning Logic's Instruments

A number of the instruments that come with Logic Pro support polyphonic pitch bend and can be retuned using the Poly MPE **Voice Mode (page 15)** in MTS-ESP MIDI Client. To enable polyphonic pitch bend click the small triangle at the bottom left of the Logic instrument UI to show the MIDI settings and set MIDI Mono Mode to 'On (with common base channel 1)' and Mono Pitch Bend Range to '48 semi'. Next add an instance of MTS-ESP MIDI Client to the track and select the 'MIDI Poly MPE' preset from the **preset menu (page 12)**.
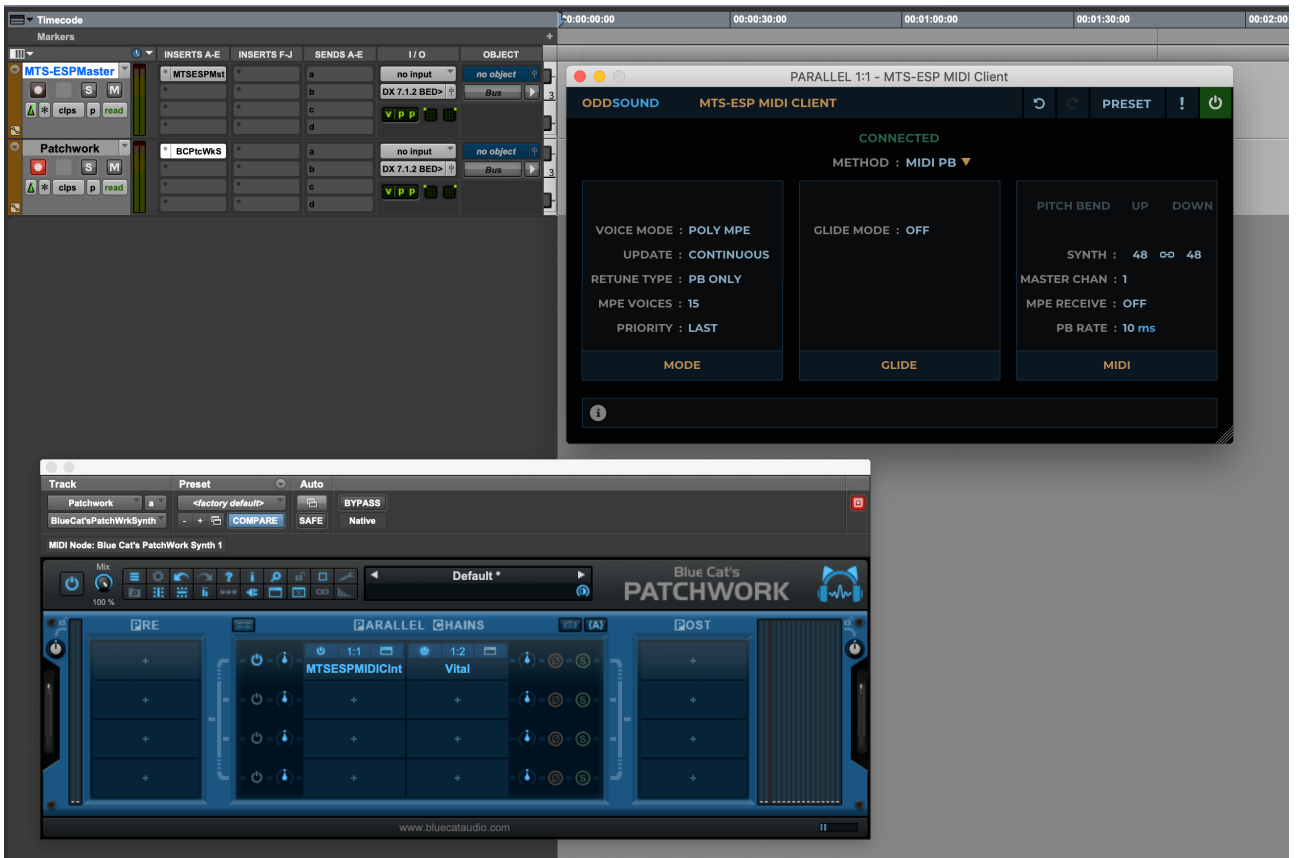
To retune a virtual instrument in Pro Tools, create two instrument tracks, one with an instance of MTS-ESP MIDI Client and another with an instance of a virtual instrument. Note that this method is limited to monophonic retuning as Pro Tools can only transfer MIDI data between tracks on a single MIDI channel, so the 'MIDI Mono' preset is a good starting point for setting up MTS-ESP MIDI Client.

Open the Mix window and select View->Mix Window Views->Instruments to see the MIDI routing options. Set the input for the virtual instrument track to 'MTS-ESP MIDI Client MIDI Out->all channels' and record-enable it.

To retune a hardware instrument, create a MIDI track and set it's input in the same way, again making sure to record-enable it.
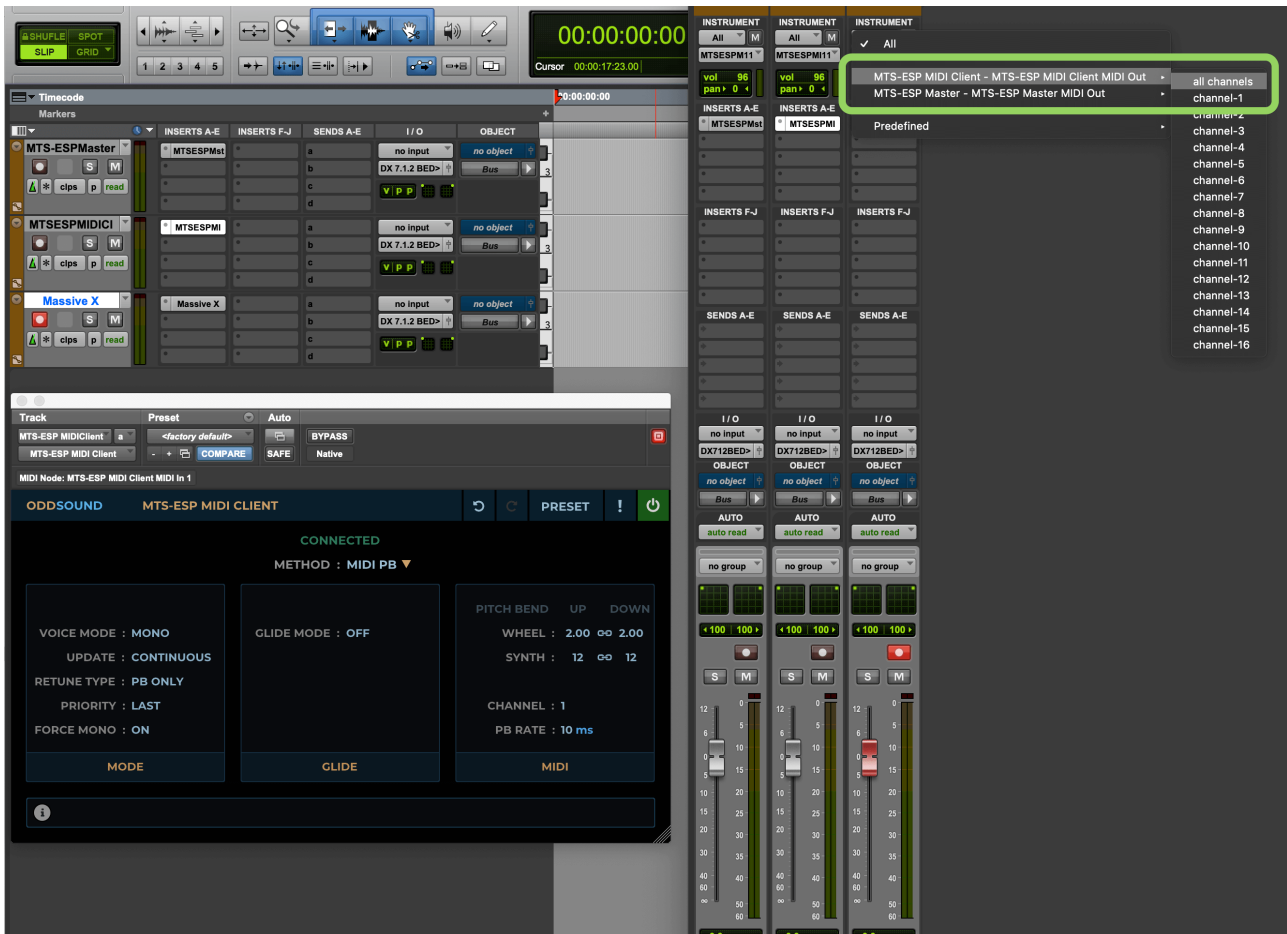
To use polyphonic pitch bend or MPE to retune a virtual instrument in Pro Tools we recommend using a hoster plug-in that can load both MIDI and instrument plug-ins. See the end of this manual from some recommended **hoster plug-ins (page 54)**.

To retune a virtual instrument in Reaper simply insert an instance of MTS-ESP MIDI Client before it in the channel strip. To retune a hardware instrument just make sure it is inserted on the track that is sending MIDI to the device.
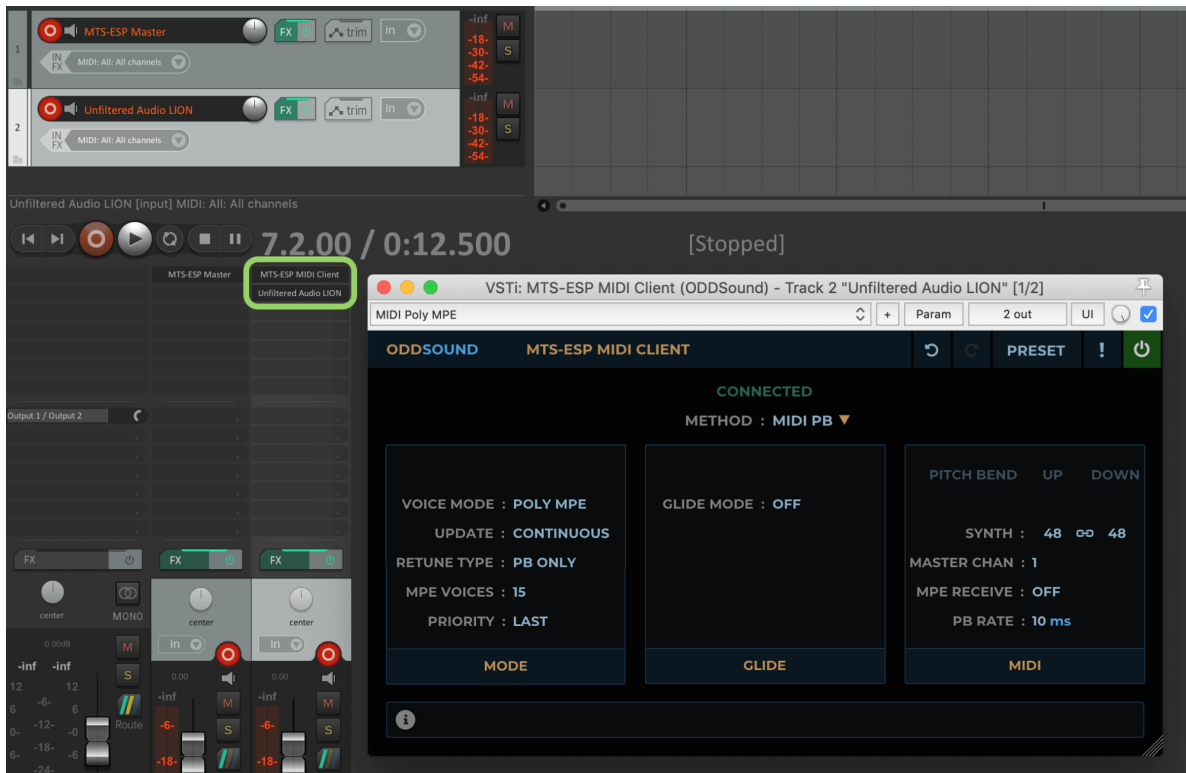
To retune a virtual instrument in Renoise, add an instance of MTS-ESP MIDI Client and an instance of the virtual instrument to be retuned. Select MTS-ESP MIDI Client and set the MIDI routing to the virtual instrument.

Note that this method can only be used for monophonic retuning since Renoise only allows a single MIDI channel to be selected at the input to a virtual instrument.

For polyphonic retuning, or even monophonic retuning that requires no routing set up, we recommend using a hoster plug-in which allows a MIDI plug-in and a virtual instrument to be loaded together inside a single plug-in. See the end of this manual from some recommended **hoster plug-ins (page 54)**.
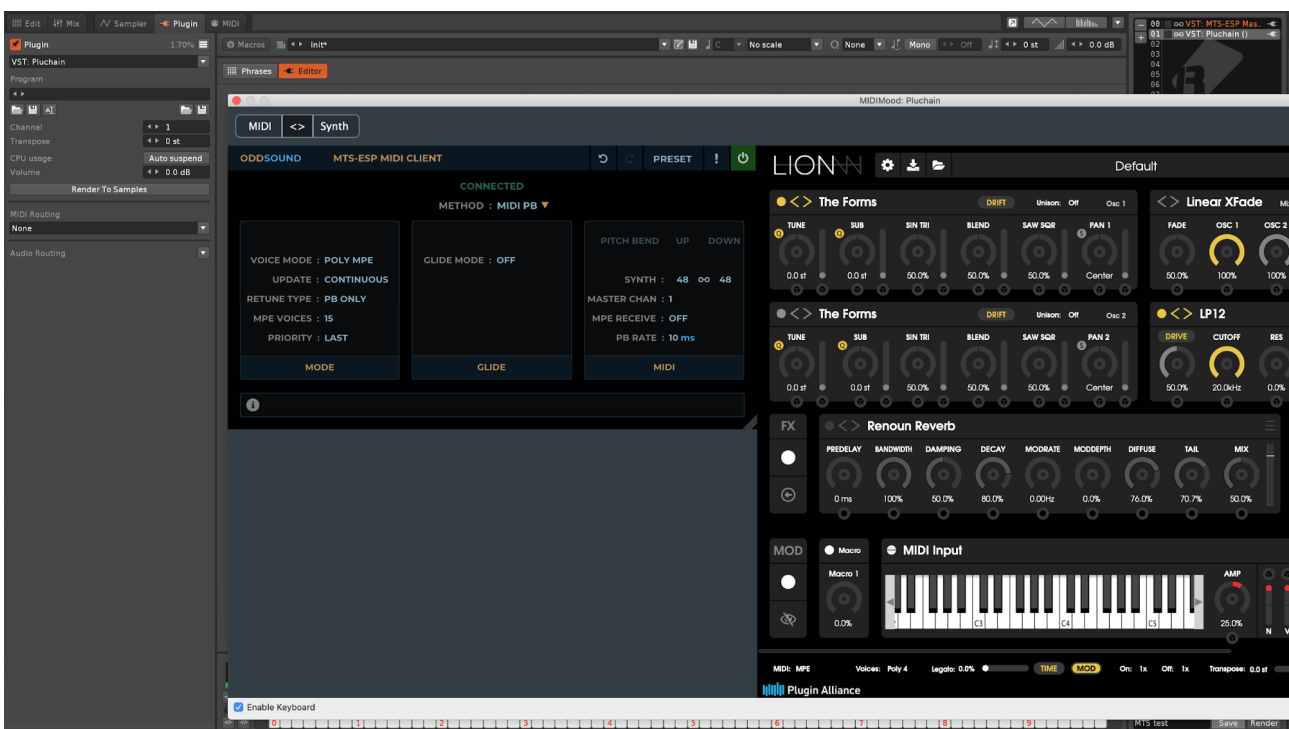
Set up steps for the MIDI Client in Studio One differ according to the selected **Voice Mode (page 15)**, plug-in format and capabilities of the synth being retuned.

Monophonic Pitch Bend Retuning With A Virtual Instrument

These steps apply to monophonic retuning i.e. parts where only a single note plays at a time:

- Create an instrument track and assign it to an instance of MTS-ESP MIDI Client. The VST2 version of MTS-ESP MIDI Client should be preferred here.
- Select the 'MIDI Mono' preset in MTS-ESP MIDI Client.
- Add a virtual instrument, but don't create a track for it. Set its Direct MIDI Input to MTS-ESP MIDI Client. Make sure that the direct input channel matches the **Channel (page 17)** setting in MTS-ESP MIDI Client.
- Make sure the **Synth Bend Range (page 17)** in MTS-ESP MIDI Client matches that of the synth being retuned.
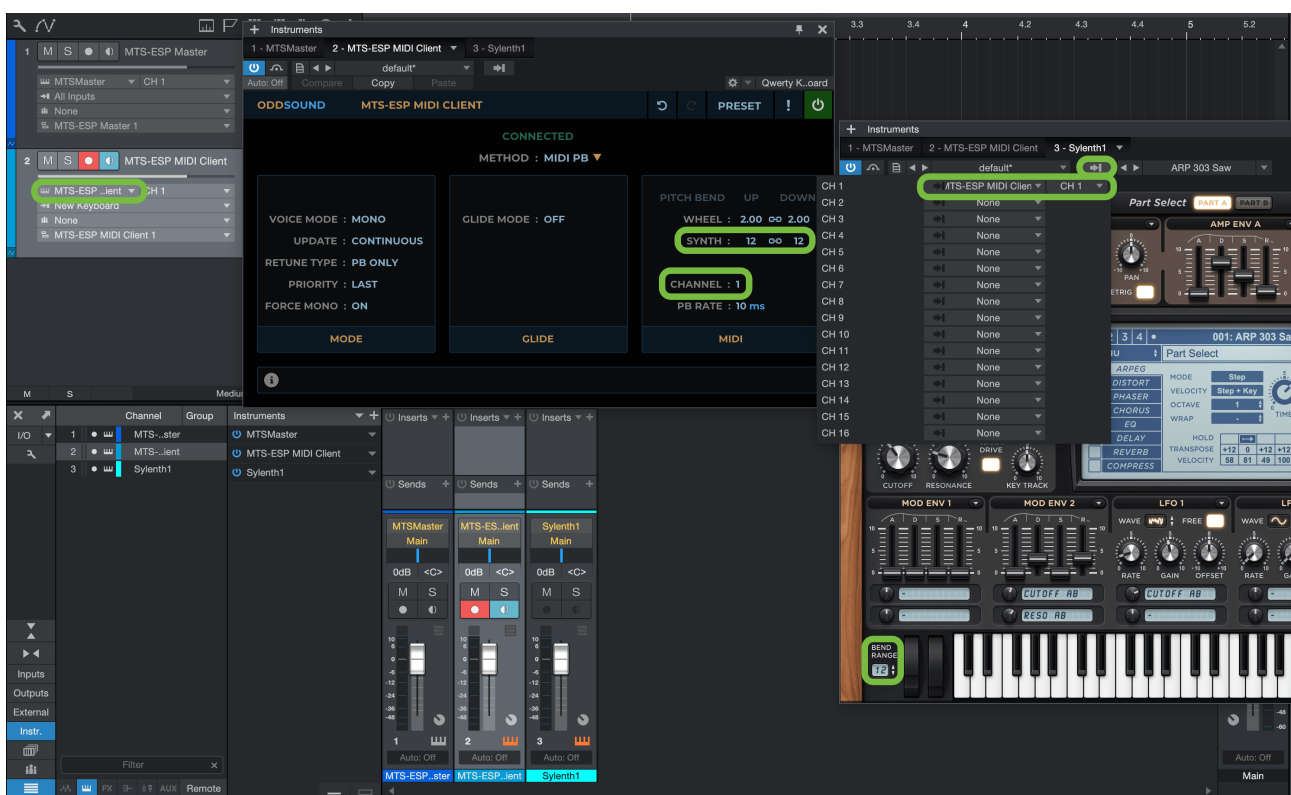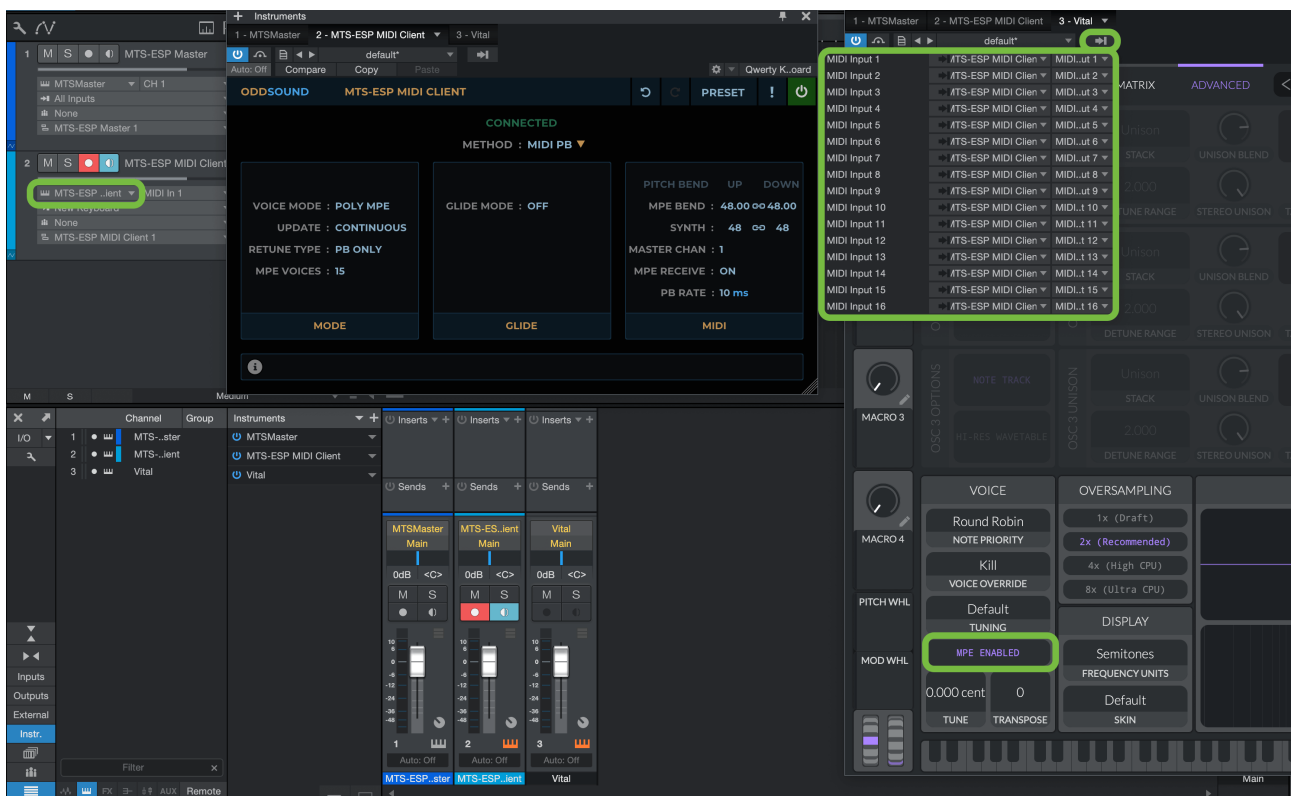

Polyphonic Pitch Bend Retuning With A VST3 Virtual Instrument Using MPE (Studio One 5.3+)

Studio One supports MPE and it is possible to use MTS-ESP MIDI Client for polyphonic pitch bend retuning in conjunction with this (including using an MPE controller), but only with the VST3 version and with a VST3 format virtual instrument that supports MPE which can accept multi-channel direct MIDI input.

- Create an instrument track and assign it to a VST3 instance of MTS-ESP MIDI Client.
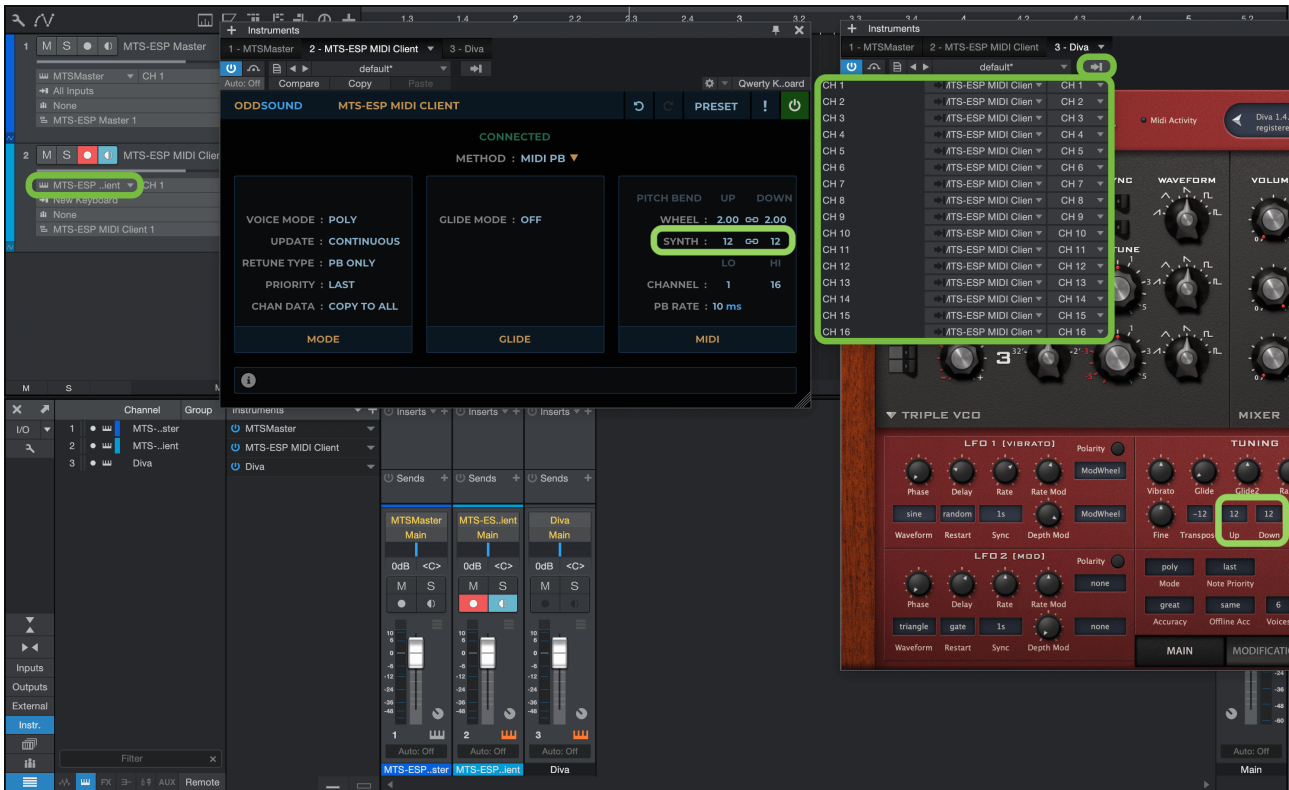- Enable MPE for MTS-ESP MIDI Client in Studio One.
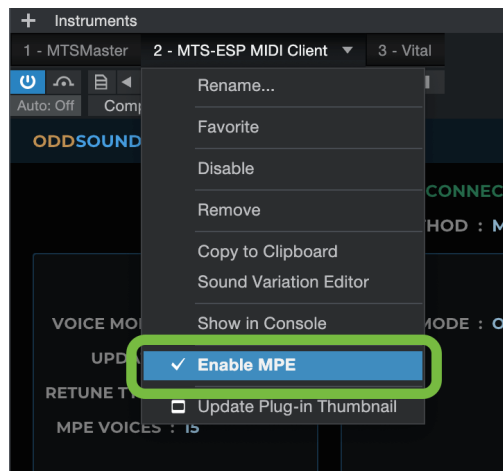
- Select the 'MIDI Poly MPE Receive' preset in MTS-ESP MIDI Client.
- Add a VST3 virtual instrument that supports MPE, but don't create a track for it.
- Enable MPE for the virtual instrument in Studio One, and also in the plug-in itself if it has an option for it.
- Set MTS-ESP MIDI Client as the direct MIDI input for all 16 channels of the virtual instrument. Make sure each input channel is set to the corresponding output channel of MTS-ESP MIDI Client.

Polyphonic Pitch Bend Retuning With A Virtual Instrument

These steps apply for polyphonic pitch bend retuning where a virtual instrument supports voice-per-channel, or where it supports MPE but is not VST3 format. Note that in the latter case it will not be possible to make use of Studio One's MPE support or use an MPE controller.

- Create an instrument track and assign it to an instance of MTS-ESP MIDI Client. The VST2 version of MTS-ESP MIDI Client should be used here.
- If retuning a virtual instrument that supports MPE, select the 'MIDI Poly MPE' preset in MTS-ESP MIDI Client and, if the plug-in has an option to enable MPE support, switch it on but do not enable MPE in Studio One. If retuning a virtual instrument that supports voice-per-channel use the 'MIDI Poly' preset, making sure that the **Synth Bend Range (page 17)** in MTS-ESP MIDI Client matches that of the synth being retuned.
- Add a virtual instrument that supports MPE or voice-per-channel, but don't create a track for it.
- Set MTS-ESP MIDI Client as the direct MIDI input for all 16 channels of the virtual instrument. Make sure each input channel is set to the corresponding output channel of MTS-ESP MIDI Client.

Using MTS SysEx Mode In Studio One

Studio One supports SysEx data. To send MTS SysEx to a hardware synth use a second instrument track to route MIDI data from MTS-ESP MIDI Client to a hardware instrument device.

To retune a virtual instrument in Waveform, simply add an instance of MTS-ESP MIDI Client on the same track as the virtual instrument, right before it. Set up MTS-ESP MIDI Client as required for the instrument being retuned. The VST2 version of MTS-ESP MIDI Client should be preferred here.
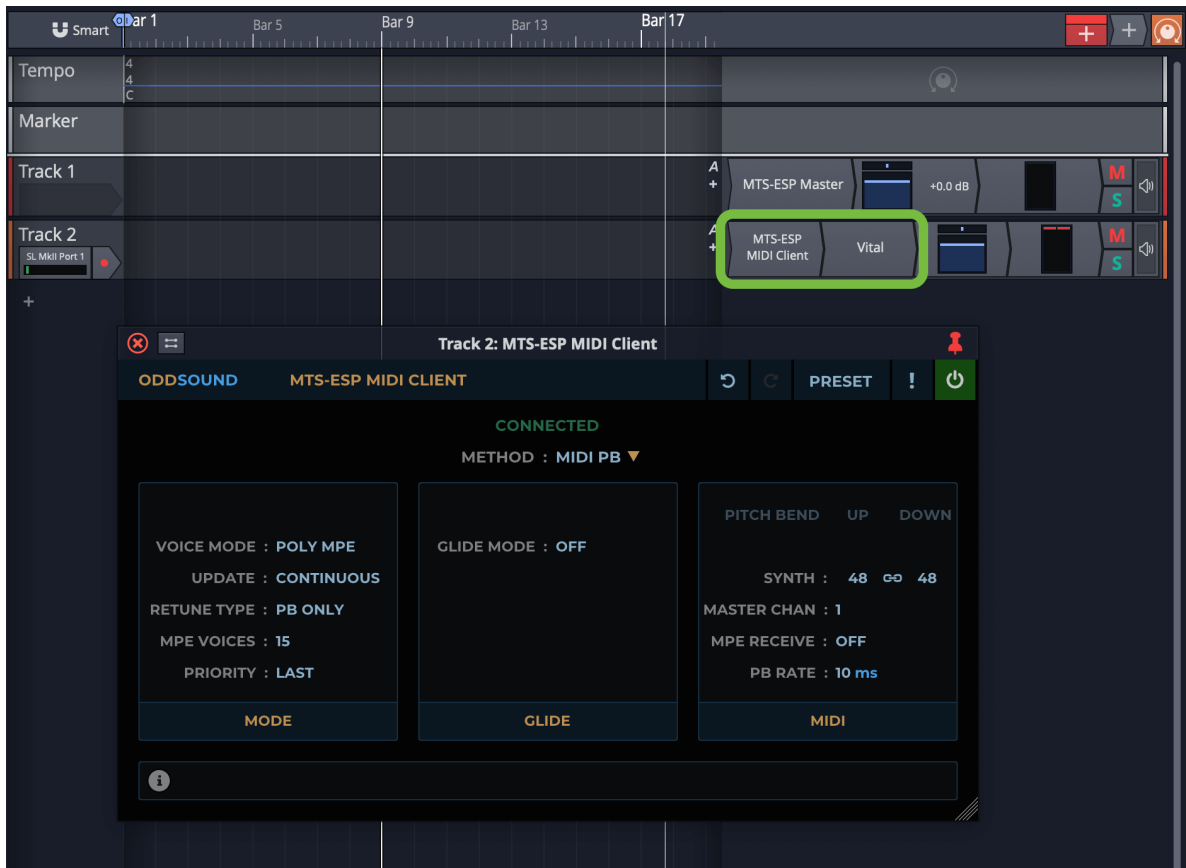
A number of the specific DAW instructions in this manual advise using a hoster plug-in in which to load MTS-ESP MIDI Client along with the virtual instrument to be retuned. Here is a list of some hoster plug-ins that may be used for this:

- Blue Cat Audio: PatchWork (**www.bluecataudio.com/Products/Product_PatchWork**)
- Kushview: Element (**kushview.net/element**)
- MIDIMood: Pluchain (**www.midimood.com/product/pluchain**)
- New Sonic Arts: Freestyle (**www.newsonicarts.com/html/freestyle**)
- Plogue: Bidule (**www.plogue.com/products/bidule**)