



d16 group

Syntorus 1.0.0

/sintorʊs/

Owner's manual

Acknowledgement: Sebastian Bachliński
Gary Brennan

February 19, 2010

Chapter 1

Overview

Syntorus is a chorus effect emulator with double delay line.

1.1 Interface

After loading the plug-in to a host application the GUI appears:

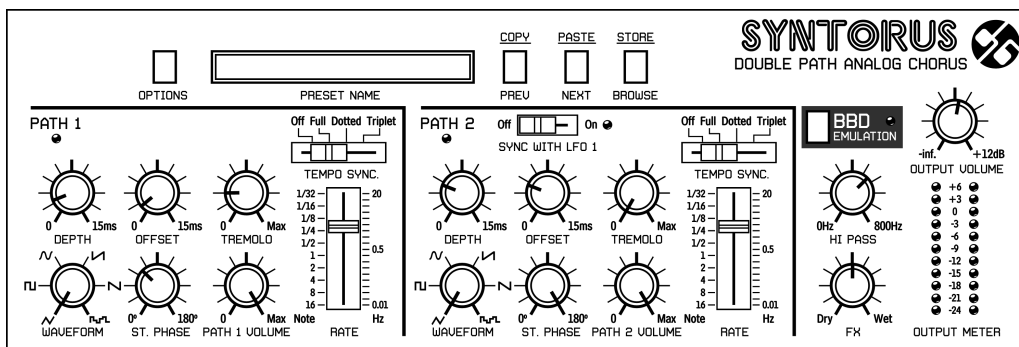


Figure 1.1: Syntorus graphical interface

We can distinguish two sections there:

- Configuration and preset management

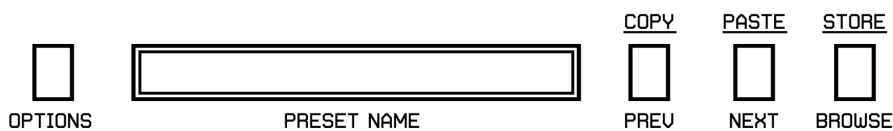


Figure 1.2: Configuration and preset management section

- Signal processing control section consists of the all remaining controls.

Chapter 2

Signal flow

This chapter describes the path of the signal's flow through the Syntorus plug-in, presents the basic components of this effect unit and its control parameters.

2.1 Basic modules

The inside of Syntorus consist of a few basic components, that correspond to the sections on the graphical interface:

2.1.1 Path 1

It's a first of two delaying lines in Syntorus controlled by its own **LFO**, which can be synchronized to the host application. Apart from that each **Path** can work as **Tremolo** as well.

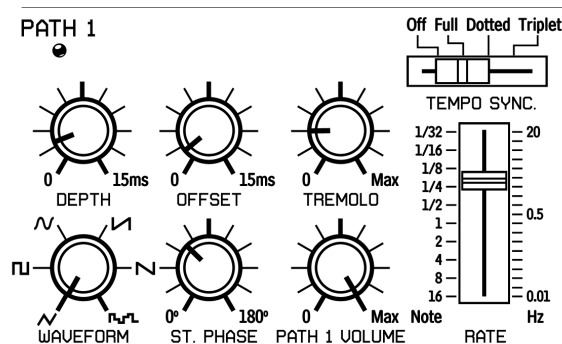


Figure 2.1: First path

Following parameters control a first path:

Waveform - Chooses a shape of **LFO**'s waveform among: triangle, square, sine, sawtooth (ascending or descending), noise.

Offset - It's a offset between dry signal and the minimum of the **LFO**'s oscillations expressed in milliseconds.

Depth - Amplitude of **LFO**'s oscillations expressed in milliseconds.

Rate - Controls **LFO**'s frequency.

To see how the **LFO** controls a delay line take a look at the figure below:

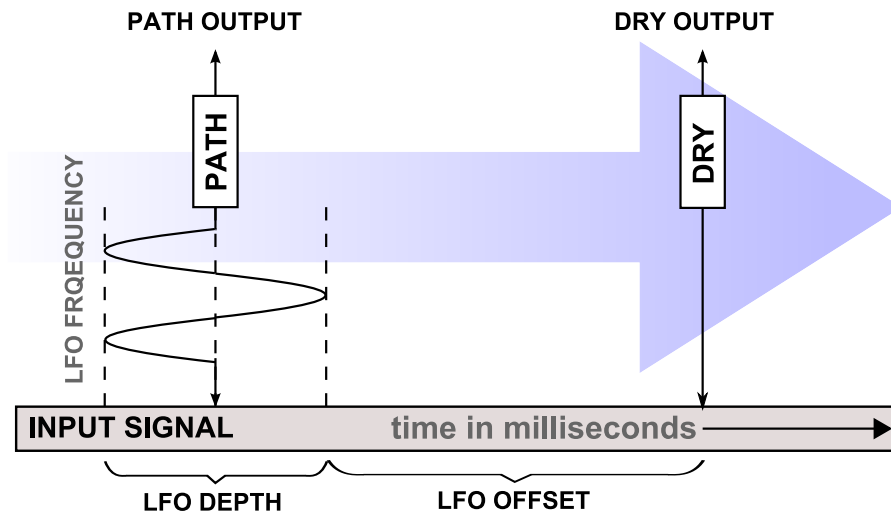


Figure 2.2: Single delay line

If You imagine a **Path** (delay line) as a reading head moving through signal, in that case **LFO** controls its acceleration.

Path volume - **Path**'s output volume.

Tremolo - Adds a tremolo effect to the **Path**'s output, frequency of tremolo is also controlled by a **Rate** knob.

St.Phase - In a case of stereo signals, left and right channel are processed independently. That means, left and right channel has its own **LFO**. When **St.Phase** is set to 0 degree, **LFO** for right and **LFO** for left channel have the same phase. Increasing **Phase** value, increases the phase shift between **LFOs**, making a pleasant stereo effect.

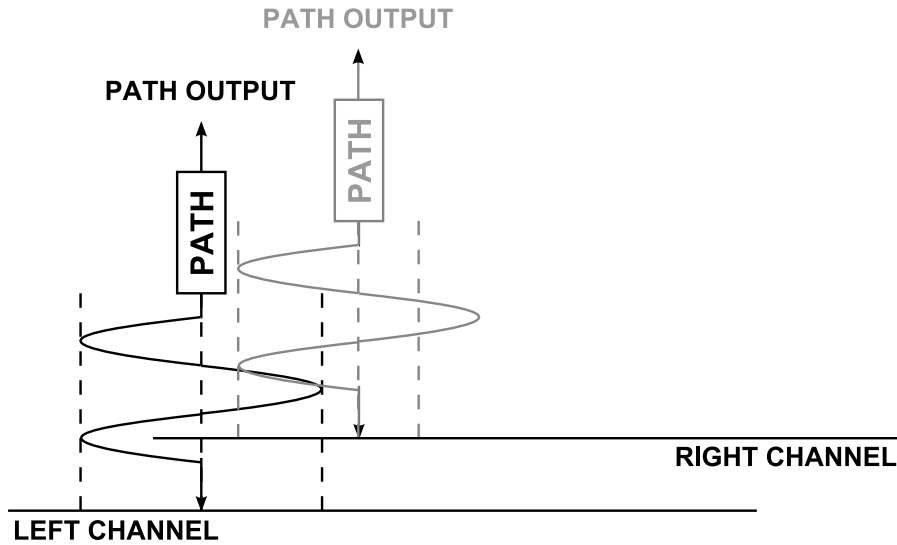


Figure 2.3: LFO stereo phase shift

When Syntorus is inserted in a mono track, **St.Phase** knob doesn't work.

Temp sync. - It's possible to sync **Path's LFO** with host application. This switch is used for this purpose. There're four available values:

- **Off** - Synchronization is off, the **LFO's** frequency is controlled via **Rate** fader arbitrarily in a range 0 to 20 Hz.
- **Full** - Synchronization is on, the **LFO's** period length is equal to a single note set on **Rate** fader, respectively to host's tempo value.
- **Dotted** - Synchronization is on, the **LFO's** period length is equal to a single dotted note set on **Rate** fader, respectively to host's tempo value.
- **Triplet** - Synchronization is on, the **LFO's** period length is equal to a single triplet note set on **RATE** fader, respectively to host's tempo value.

When synchronization is on, **LFO's** rate can be set to following values: 1, 2, 4, 8 and 16 bars, half note ($\frac{1}{2}$), quarter note ($\frac{1}{4}$), eighth ($\frac{1}{8}$), sixteenth ($\frac{1}{16}$) and thirty-second ($\frac{1}{32}$).



Figure 2.4: One bar

Dotted note is a $\frac{3}{2}$ longer than **Full** note, **Triplet** note is $\frac{2}{3}$ of **Full** note.

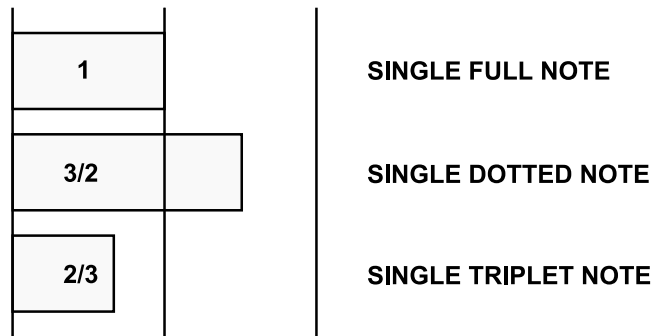


Figure 2.5: Note lengths

2.1.2 Path 2

Basically **Path 2** doesn't differ to a **Path 1**. Using two **Paths** at once makes sound even fatter. The only difference is a **Sync with LFO 1** switch. When this switch is set to **Off** position, the two Paths process signal totally independently. But when we set that switch to **On** position, the second path (**Path 2**), becomes a slave. All of its controls work in regular way, apart from a **Rate** fader, which value is taken from **Rate** fader of **Path 1** - that includes also synchronization.

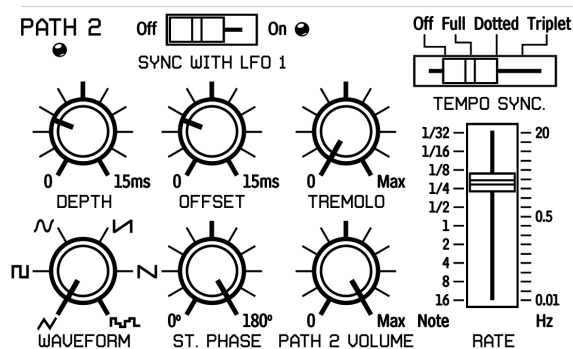


Figure 2.6: Second path

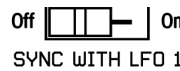


Figure 2.7: Sync with LFO 1 switch

2.1.3 Master section

Controls the end of signal's flow.

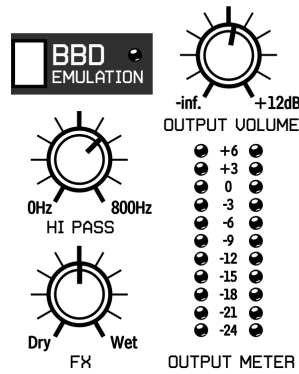


Figure 2.8: Master section

Dry/Wet - Controls proportions between unprocessed and processed signal sent to output.

Output volume - Controls the final amplification.

BBD - It's a switch that turns on **Bucket-Brigade Device** emulation, which is discrete-time analogue delay line. It causes bigger usage of CPU, but affect on more pleasant and warm character of chorus.

Hi pass - There's a hi-pass, that filters whole wet signal outgoing from delay lines. This parameter controls filter's cut off frequency in a range of $[0Hz \dots 800Hz]$.

2.2 Path of the signal's flow

The picture below shows how the signal flows through the plug-in:

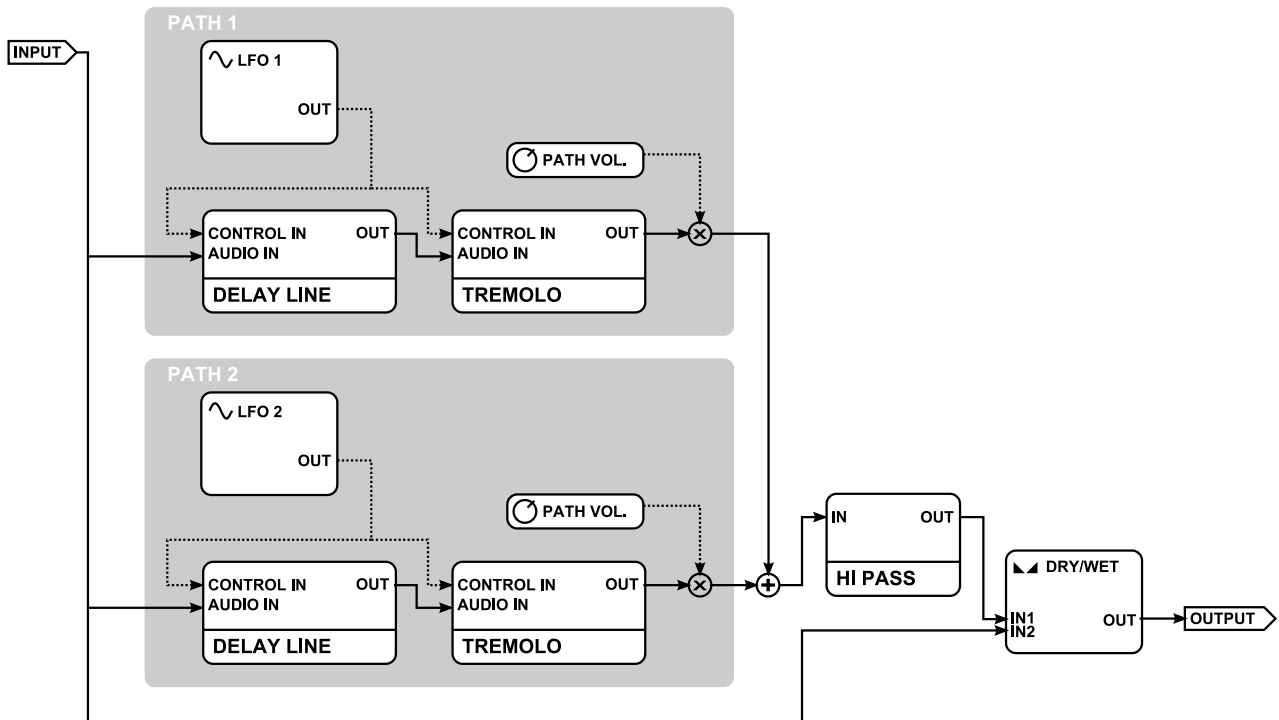


Figure 2.9: Signal flow

Chapter 3

Preset management

3.1 Browsing presets

Presets are organized into groups. Storing like this is not compatible with the native method used by the host application. The user can see presets in the browser as being assigned to particular groups (defined by user).

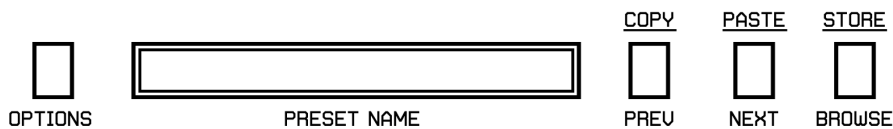


Figure 3.1: Configuration and preset management section

The user interface Preset Management controls are:

Preset name - This displays the name of currently selected preset. It also allows editing of the preset name.

Prev/Next - Those buttons are used to navigate through all presets (whole bank), **Next** button moves to the next slot, and **Prev** button to the previous one. When the end of a group of presets is reached, the first slot of the next group will be chosen when the **Next** button is pressed again. Similar action will be performed if we press the **Prev** button when the beginning of a group is reached.

Prev + Ctrl - **Prev** button pressed with **Ctrl**¹ key, copies current preset slot to buffer.

Next + Ctrl - **Next** button pressed with **Ctrl** key, pastes buffer to current preset slot with postfix *_copy* added to its name.

Browser - Allows selection of presets using a browser menu.

¹On MacOS use **Apple** key instead of **Ctrl** key.

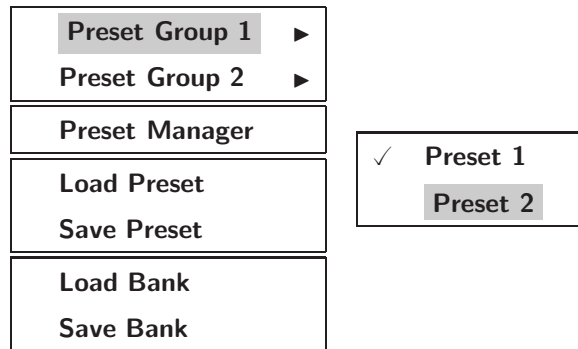


Figure 3.2: Browsing presets via context menu

It is worth mentioning that changing any of the presets is not permanent. After removing and reloading the plug-in again, the factory presets will be recalled. However, saving the project within the host application with Syntorus, and reloading that project will cause a recall of all changes we have made in this project.

3.2 Loading and Saving presets

There are some additional options that add functionality to preset management. They are placed in the context menu which is accessible from the **Browse** button:

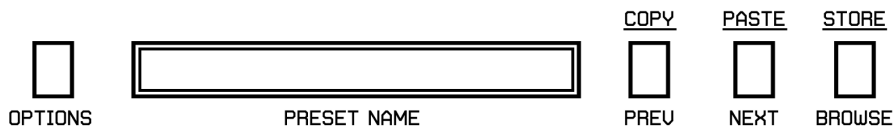


Figure 3.3: Configuration and preset management section

When we click this button, the context menu is unfolded:

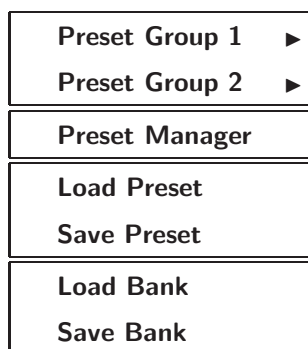


Figure 3.4: Preset management - under Browse button

In this chapter, we'll focus on the first 4 items:

Load Preset - Loading preset and overwrite the active slot from the file (**.snprs- Syntorus Preset**).

Save Preset - Saving current preset as a file (**.snprs- Syntorus Preset**) ².

Load Bank - Loading whole bank of presets from file (**.snprsb- Syntorus Bank**).

Save Bank - Saving whole bank of presets as a file (**.snprsb- Syntorus Bank**).

Note: *Internal files written by a Syntorus are in XML format and can be edited in a text editor.*

3.3 Preset manager

As was mentioned earlier, presets in Syntorus are organized into groups. Presets can be chosen from the context menu (under the **Browse** button). **Preset manager** is a tool, which allows to easy management of the presets structure. To open it, just click the **Browse** button:

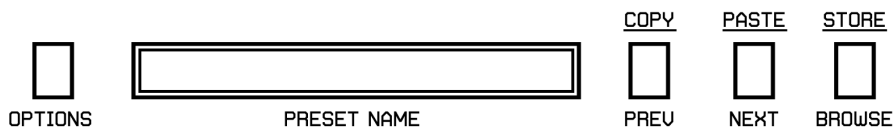


Figure 3.5: Configuration and preset management section

From the context menu choose **Preset Manager**:

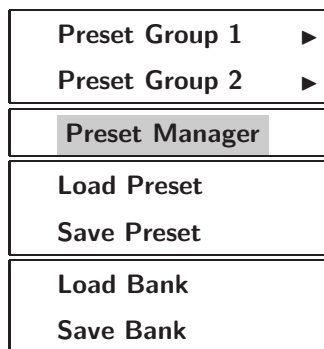


Figure 3.6: Preset manager - Item

The Preset Manager menu will appear.

²Before saving single preset, remember to **Store** it using **Ctrl + Browse** button if **On demand** preset storing is active, which is default behavior.

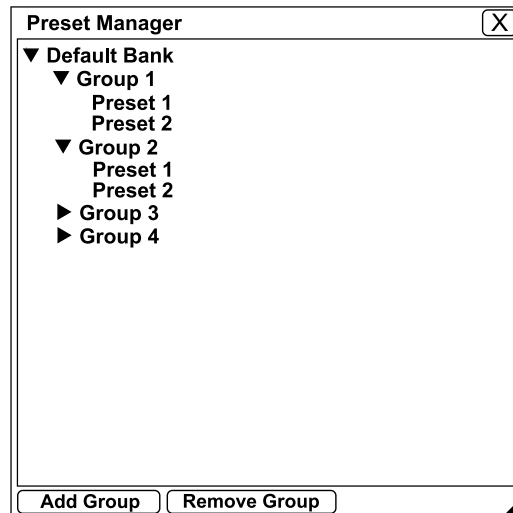


Figure 3.7: Preset manager window

This shows the preset structure with the bank name as the root. A preset represents the deepest level of the structure.

Each of the items can be renamed by double clicking on the item with the mouse.

The preset structure appears as a standard list of presets within the host application making it fully compatible with it. As a result, no matter how the structure is represented (either as a tree structure or flat), the maximum number of presets is 128. The number of presets in this structure cannot be changed. We can only move the presets from one group to another (by dragging) or changing their order in group. The Group order in the bank can also be changed by dragging.

There are two buttons at the bottom of the **Preset Manager** window.

Add group - Adding the empty group in a bank.

Remove group - This removes a group from a bank – but only when it is empty. It means that all preset slots were reallocated among the existing groups.

To exit from the **Preset manager** press the **Cross** button in the top right corner.

3.4 Preset storing

Using **Next**, **Prev** buttons within **Preset Manager** we can navigate through the preset bank. Any change made in preset can be stored automatically or on demand:

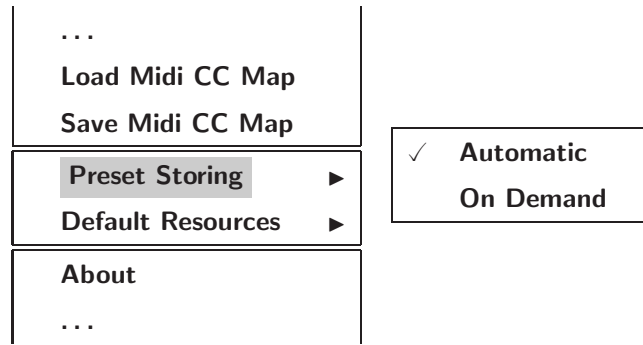


Figure 3.8: Preset storing flag

Automatic - When any parameter in the current preset is changed, it is automatically stored.

On demand - If a parameter is changed, the change is not saved within the preset until the **Store** button is pressed (**Ctrl** key + **Browse** button)³. Navigating to a new preset will cause changes to be lost unless stored. This is the default.

³On MacOS use **Apple** key instead of **Ctrl** key

Chapter 4

Configuration

4.1 Midi control

Syntorus has an ability to assign its controls (on GUI) to any **Midi Control Change** code (**MidiCC**), This allows control of the plug-in using external software or hardware.

Note: *This feature works only in VST version, AU effect plug-in has no midi input necessary to receive midi messages.*

4.1.1 Midi learn

To assign a Syntorus control to a midi controller:

1. From the context menu under **Options** button, we set **Midi Learn Mode**.

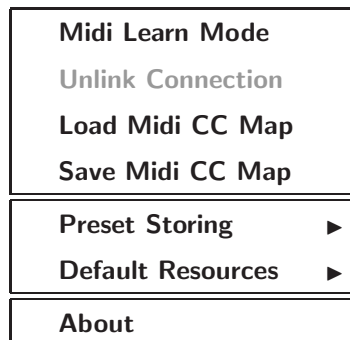


Figure 4.1: Options Menu

2. In the **Preset name** edit box, the *waiting...* message will appear. In this mode, plug-in is waiting for any movement of its controls on GUI or any of controls on the external midi controller connected to a host (with an active midi input channel directed to Syntorus). The edit box will display the name of currently modified control.
3. When the control is set, go to the **Options** context menu and uncheck the **Midi Learn Mode** option. Last modified control on GUI will be assigned to the last moved control on the midi controller.

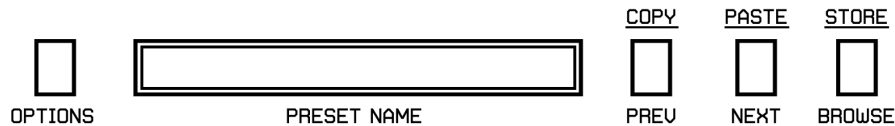


Figure 4.2: Configuration and preset management section

4.1.2 Midi Unlink

To disconnect midi controller and Syntorus GUI's control:

1. From the context menu under **Options** button, we set **Midi Learn Mode** first.

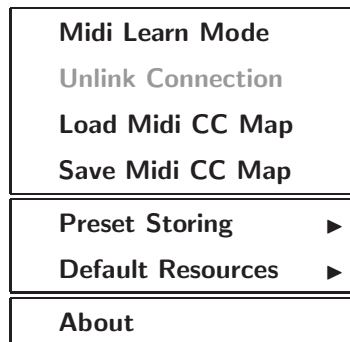


Figure 4.3: Options Menu

2. In the **Preset name** edit box, the *waiting...* message will appear. In this mode, plug-in is waiting for any movement of its controls on GUI or any of controls on the external midi controller connected to a host (with an active midi input channel directed to Syntorus). The edit box will display the name of currently modified control.
3. **Unlink Connection** item in **Options** menu is activated now. Click it to disconnect Midi CC and GUI's control. ¹
4. Uncheck the **Midi Learn Mode** option.

4.1.3 Loading and saving Midi Map

To save a Midi Map (mapping of GUI's controls with MidiCC), we can use the option from the context menu, which is accessible under the **Options** button:

¹You're allowed to unlink few connections one by one, by repeating steps 2 and 3.

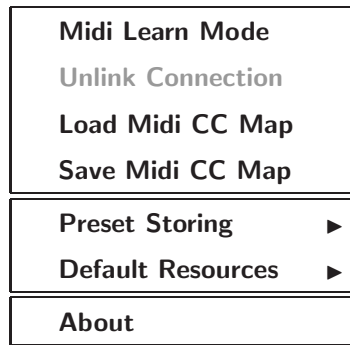


Figure 4.4: Options Menu

Load Midi CC Map - Loading a Midi Map from file (**.snccmap- Syntorus Control Map**).

Save Midi CC Map - Saving a current Midi Map to file (**.snccmap- Syntorus Control Map**).

Note: *Midi Map files written by a Syntorus are XML standardized, it means You can re-edit them within any text editor.*

4.2 Default resources

There're included some default presets with Syntorus or empty Midi CC map. These are loaded every time it is started. This can be changed to point to user default presets or default Midi CC map.

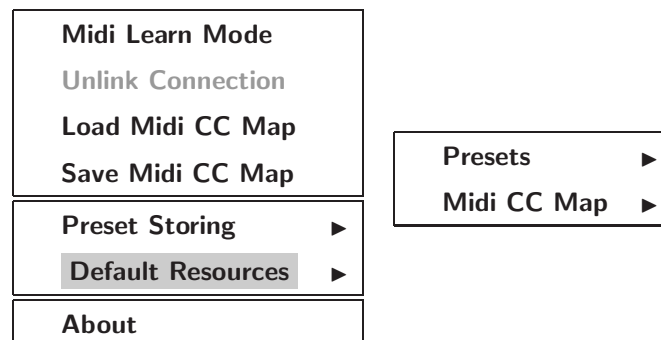


Figure 4.5: Default resources

The following types of resources can be replaced by user's ones and those will be loaded every time new instance of plug-in be loaded:

Presets - Default Syntorus's Preset Bank (**.snprsb**)

Midi CC Map - Default Syntorus's Midi CC Map (**.snccmap**)

4.2.1 Default presets

To set the default bank of presets, use the options from the **Presets** submenu:

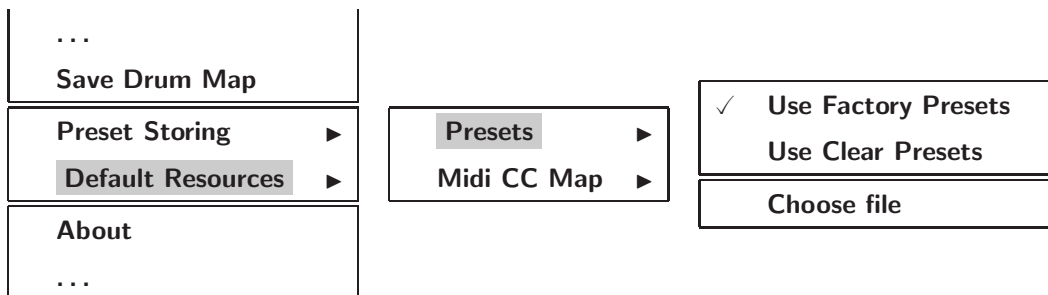


Figure 4.6: Default presets

We can choose one of following options:

Use Factory Presets - This is default one after installation of Syntorus. If You choose this option, default bank of presets becomes the factory one provided within plug-in by producer.

Use Clear Presets - After choosing this item from menu, the default bank will contain all presets consisting of zeroed parameters.

Choose File - It's possible to set by user his own default bank of presets he previously made, by using this option. **Choose File** opens a file dialog to browse a location where the bank of presets made by user is saved. When user confirms, the path to that bank is stored in the Syntorus's configuration file. And this bank becomes the default one.

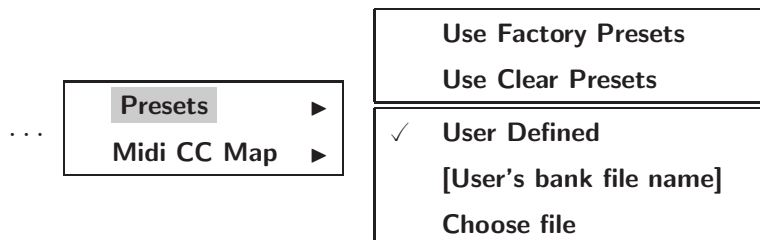


Figure 4.7: User's default presets

When the user's bank is set as a default preset bank, the menu indicates this fact by showing the checked **User Defined** menu item, and one row below the file name of user's preset bank.

4.2.2 Default Midi CC Map

The option used to set the default Midi CC map.

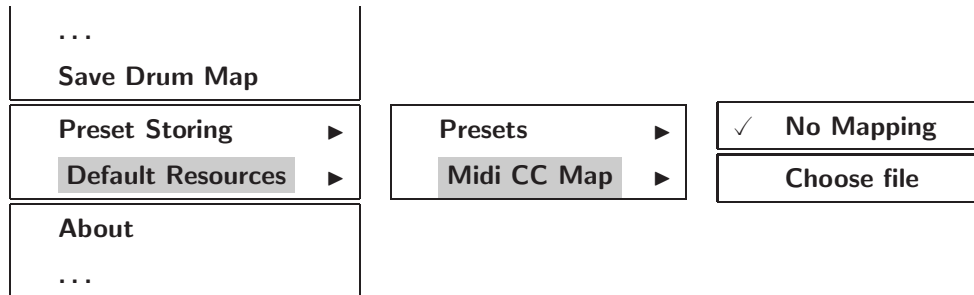


Figure 4.8: Default Midi CC Map

Only two options are available in this submenu:

No Mapping - It is the default one, mean no default Midi CC mapping is available after loading Syntorus into the host application.

Choose file - Similarly to selecting default preset resource, this options is used to set the default Midi CC Map, which will be loaded every time we load Syntorus in the host application. When we set the path to Midi CC Map using this options, it's indicated by showing Midi CC Map file name in this submenu:

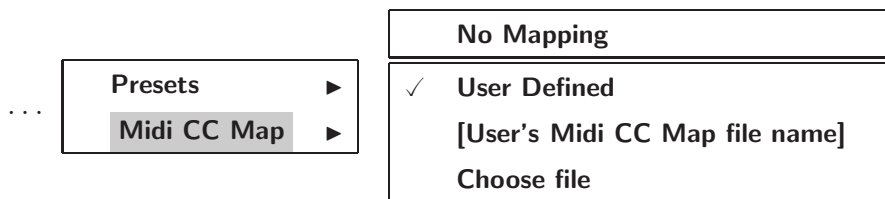


Figure 4.9: User's default Midi CC Map

Chapter 5

Contributions

We'd like to thank the people who helped us with Syntorus' development. For beta testing and factory presets:

- Clifford Douse
- Colin Stark
- Kasper Toeplitz
- Laurent De Fru aka Laurent Bergman

For providing us samples from classical hardware units:

- Kristian Per

Contents

1	Overview	1
1.1	Interface	1
2	Signal flow	2
2.1	Basic modules	2
2.1.1	Path 1	2
2.1.2	Path 2	5
2.1.3	Master section	5
2.2	Path of the signal's flow	7
3	Preset management	8
3.1	Browsing presets	8
3.2	Loading and Saving presets	9
3.3	Preset manager	10
3.4	Preset storing	11
4	Configuration	13
4.1	Midi control	13
4.1.1	Midi learn	13
4.1.2	Midi Unlink	14
4.1.3	Loading and saving Midi Map	14
4.2	Default resources	15
4.2.1	Default presets	16
4.2.2	Default Midi CC Map	16
5	Contributions	18



SYNTHORUS

DOUBLE PATH ANALOG CHORUS

COPY PASTE STORE
 PREVIOUS NEXT BROWSE

PRESET NAME

OPTIONS

PATH 1

Off Full Dotted Triplet

TEMPO SYNC. 20 Hz

1/32 1/16 1/8 1/4 1/2 1 2 4 8 16

DEPTH 0 15ms

TREMOLO 0 Max

OFFSET 0 15ms

ST. PHASE 0° 180°

WAVEFORM

Max Note

PATH 1 VOLUME

PATH 2

Off Full Dotted Triplet

TEMPO SYNC. 20 Hz

1/32 1/16 1/8 1/4 1/2 1 2 4 8 16

DEPTH 0 15ms

TREMOLO 0 Max

OFFSET 0 15ms

ST. PHASE 0° 180°

WAVEFORM

Max Note

PATH 2 VOLUME

BBD EMULATION

0Hz 800Hz

HI PASS

Dry Wet

FX

OUTPUT VOLUME

+12dB

-inf.

+6 +3 0 -3 -6 -9 -12 -15 -18 -21 -24

OUTPUT METER