



# Cthulhu

The Chord and Arp Monster

VST MIDI-FX Plug-In

Version 1.1

October 2015

© 2013-2015 Xfer Records

[www.xferrecords.com](http://www.xferrecords.com)

**Cthulhu VST plug-in  
for Windows and Macintosh OS X**

Coding: Steve Duda ([www.steveduda.com](http://www.steveduda.com))

Manual: Steve Duda  
*thanks to stevejay for the manual template*

Manual version: Tuesday, September 22, 2015

*The information contained in this document is subject to change without notice. In no event shall Xfer Records or the author of this document be liable for any damages arising out of or related to this document or the information contained within it. No part of this document may be reproduced or transmitted in any form or for any purpose without the express written consent of Xfer Records.*

**VST Plugin technology by Steinberg**

VST is a trademark of Steinberg Media Technologies GmbH  
All other copyrighted trademarks belong to their respective owners.

**W W W . X F E R R E C O R D S . C O M**

# 1. Contents

---

<b>2.Introduction.....</b>	<b>5</b>
Getting in touch.....	5
Installing Cthulhu.....	5
<i>Installing on the PC.....</i>	<i>5</i>
<i>Installing on Macintosh OS X.....</i>	<i>5</i>
<i>Uninstalling Cthulhu.....</i>	<i>6</i>
<b>3.Connecting Cthulhu (VST version).....</b>	<b>7</b>
Step 1: Add Cthulhu .....	7
Step 2: Add an Instrument.....	7
Step 3: Connect them together.....	7
<b>4.Connecting Cthulhu in Logic 9 (AudioUnits version)..</b>	<b>9</b>
<b>5.Cthulhu in Logic X.....</b>	<b>13</b>
<b>6.Operating Cthulhu.....</b>	<b>14</b>
Two Modules, One Plug-in.....	14
Mute Output (MIDI only).....	14
<b>7.Chords Module.....</b>	<b>15</b>
Chord – One note in becomes many.....	15
Altering a chord within a Chord Slot.....	16
Chord Panel Area (alphanumeric Display of Chord).....	17
Lock switches.....	17
Macro Switches (Learn / Copy / Pasta / WTF).....	17
LEARN.....	17
Learn a single chord slot.....	17
Advanced Learn (Note On Only).....	17
Copy Button.....	18
Pasta Button.....	18
WTF Button.....	18
<b>8.Chord Menu.....</b>	<b>19</b>
“Sort” Sub-Menu.....	19
<b>9.Arpeggiator.....</b>	<b>22</b>
Arpeggiator – many notes in output one-at-a-time.....	22
Graphs.....	22

---

<b>10.Arpeggiator Graphs.....</b>	<b>23</b>
Note tab.....	23
Step muting.....	24
Graph pattern length.....	25
Position Reset (down arrow).....	25
Rand Sel tab.....	25
Octave tab.....	26
Pitch tab.....	26
Vel scale tab.....	27
Gate tab .....	27
Late tab .....	27
Harmony tab.....	27
<b>11.Additional Arp Features.....</b>	<b>28</b>
Arp Patterns.....	28
Note output prevention.....	29
Clock Div.....	30
Copy/Paste/Random.....	30
<b>12.Presets.....</b>	<b>31</b>
Presets Browser / Loading Presets.....	31
Anatomy of a Chord preset.....	31
Anatomy of an Arp preset.....	31
<b>13. MIDI CC and Addendum.....</b>	<b>32</b>
This chapter documents the changes since the 1.0 release (thru 1.09).....	32

## 2. Introduction

---

Cthulhu is two MIDI FX modules combined into one plug-in: One half is a chord memorizer and player, allowing you to experiment with creating and reworking chord progressions with single-note presses. The other half is a pattern-based arpeggiator, allowing incoming notes or chords to be manipulated in various rhythmic ways.

Cthulhu runs inside your host program (e.g. Ableton Live™, Steinberg Cubase™, Image-Line FL Studio™, Energy XT™, Reaper™, Mackie Tracktion™, etc.) .

### Getting in touch

If you experience any problems while using this plug-in, or you just wish to pass on your comments regarding Cthulhu or this manual, or links to cool tunes made with Cthulhu, you can email the developer directly at: [steve@xferrecords.com](mailto:steve@xferrecords.com).

### Installing Cthulhu

Cthulhu is available in both PC and Mac OS X versions. It is designed to be used within a host audio application that supports the VST plug-in format.

- **Installing on the PC**

Unzip the .zip file. Place Cthulhu.dll in your VSTPlugins folder (or subfolder inside), as determined by your host program. **Place the “Cthulhu Presets” folder in the same location as Cthulhu.dll**

- **Installing on Macintosh OS X**

Run the Install\_Xfer\_Cthulhu.pkg contained on the disk image (dmg). This installer will automatically install:

- Cthulhu.vst to /Library/Audio/Plug-Ins/VST/
- Cthulhu.component and Cthulhu\_MFX.component to /Library/Audio/Plug-Ins/Components

**If you use Logic**, place Cthulhu.component in the Components folder of your choice, e.g. /Library/Audio/PlugIns/Components/. The AudioUnit (.component) sends MIDI out via IAC Bus (see chapter 4), so the AudioUnit is is not recommended in hosts which support both plug-in formats, such as Ableton Live.

**Place the “Cthulhu Presets” folder in /Library/Audio/Plug-Ins/VST or ~Library/Audio/Plug-Ins/VST.**

---

Cthulhu is a VST instrument plug-in, though it only outputs waves for reference - Do not expect to hear amazing sound from it! The intent is to route MIDI from Cthulhu to another software instrument (unless you really like simple sawtooth waves). If you are unsure as to how to do this, check the documentation for your host. Instructions for doing this in

some programs will follow in the next chapter.

---

- **Uninstalling Cthulhu**

(PC) delete the Cthulhu.dll and Cthulhu\_x64.dll from your VSTPlugins folders that you selected on install.

Delete Cthulhu Presets folder from /Users/(you)/My Documents/Xfer/

(OSX) move Cthulhu.vst from your VST Plugins folder and Cthulhu Presets folder to the trash. Note there are two VST Plugins folders on OSX, which aren't readily visible on OSX Lion or greater.

1) In the Finder, press command-shift-G (go to folder) and type or paste: /Library/Audio/Plug-Ins/VST/

drag Cthulhu.vst to trash

2) command-shift-G (go to folder): /Library/Audio/Plug-Ins/Components delete Cthulhu.component + Cthulhu\_MFX.component

3) command-shift-G (go to folder) /Library/Audio/Presets/Xfer Records/ delete Cthulhu Presets folder

# 3. Connecting Cthulhu (VST version)

---

## Step 1: Add Cthulhu

Cthulhu should be added to an Instrument (MIDI) Track. You need to be able to route MIDI to Cthulhu so you can have it trigger chords. This is the same procedure you would follow adding any VST Instrument in your host. If you are not sure how to do this, refer to your host documentation or send us an email.

## Step 2: Add an Instrument

Since Cthulhu doesn't make amazing sound itself (just a sawtooth tone for reference, if desired), you should add another instrument for Cthulhu to control, such as a VSTi, external MIDI synthesizer, or built-in instrument in your host. Keep in mind since Cthulhu can send chords, it is recommended (but not necessary, if using the Arp) that the instrument can be *polyphonic* (can play a chord back, not just a single note).

## Step 3: Connect them together

You need to route the MIDI From Cthulhu to the Instrument. The manner of doing this varies in different hosts. Generally this involves setting the MIDI input on the destination instrument track to the track containing Cthulhu. Here are two examples:

**MIDI Connection, Example 1: Ableton Live**



Steps 1, 2 as mentioned above. (Created two MIDI Tracks, added Cthulhu to Track 1, and an Instrument to the Track 2)

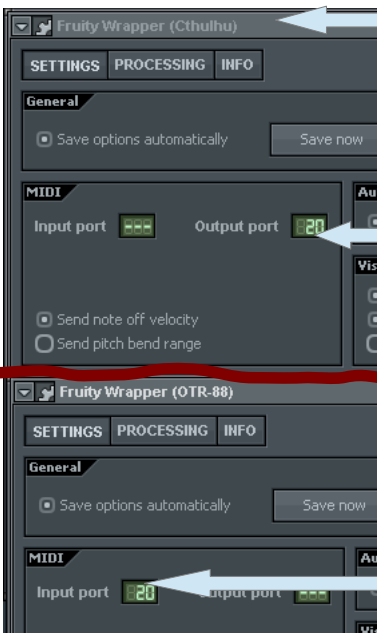
**3A)** On the track containing the Instrument, select the track containing Cthulhu as the "MIDI From" MIDI Source.

**3B)** Next, select "Cthulhu" so Live knows the MIDI is coming from Cthulhu directly.

**3C)** Lastly, set Monitor to "In", so our destination track will receive this MIDI.

Please note you will probably want to set Monitor to "In" (on the track containing Cthulhu, so you can trigger Cthulhu from your MIDI controller).

### MIDI Connection Example 2: FL Studio



**3A)** Click the Gear icon in the top-left of Cthulhu's GUI window to show the Plug-in Settings.

**3B)** Select an **Output port** that is a number greater than the Hardware MIDI Outs in the MIDI Settings of FL Studio. 20 is a safe bet, unless you are a serious hardware synth freak!

**3C)** On the Destination VSTi, show its plugin settings (Gear icon) and select the same number you chose for Cthulhu (e.g. 20) but this time in the **Input port** selector instead of Output port.

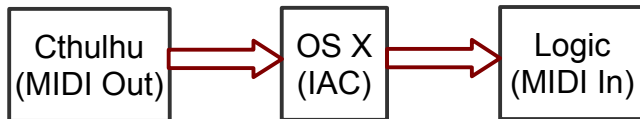


# 4. Connecting Cthulhu in Logic 9 (AudioUnits version)

---

**LOGIC X USERS:** Cthulhu will run as a MIDI FX, appearing where the Logic X Arpeggiator, etc. appears. You can consider this chapter as “Legacy” only for Logic 9 users and skip to chapter 5 (although you could use Cthulhu the way outlined here in this chapter, the method in the next chapter is much cleaner).

Due to limitations in Logic/AudioUnits under Logic 9, Cthulhu has to use the operating system in order to “Loopback” MIDI from Cthulhu to your desired destination synth:



Above is an abstract diagram of MIDI flow from Cthulhu (running inside Logic) sending MIDI to the Mac OS, which in turn routes the MIDI “back” to Logic as a MIDI input.

There are two ways to connect MIDI to Logic.

## Option 1: MIDI Out in Logic, “Secure Method”

Fortunately, the first step is a one-time procedure, and the rest of the connections could be saved as part of your Autoload template for Logic, if you so desire.

### **1. Make sure an IAC bus is installed in your system**

Open Audio MIDI Setup (Applications -> Utilities -> Audio MIDI Setup)

Double-click the IAC Driver icon in the MIDI window and enable it in the

pop-up window (“Device is online”).

## 2. Create a new Software Instrument track, and add Cthulhu to this track as an instrument

This is done the same way you would add any software instrument, Cthulhu will appear under the “AU Instruments->Xfer Records”. If Cthulhu is not showing, make sure you are selecting the instruments pop-up, not an effect insert. If it is still not visible, run Logic->Preferences->AU Manager.. and make sure it is enabled in the list (“reset and rescan selected”).

Cthulhu, when running will realize that it is running as an AudioUnit will attempt to connect to the highest-numbered IAC bus in the system.

## 3. Verify that the IAC Bus is located.

When the Cthulhu window is open, you will see the IAC Driver in the top-area which is in-use of the Cthulhu window, as pictured below:



If it instead displays “MIDI: IAC BUS NOT FOUND!!” go no further – refer to step 1 and make sure IAC is enabled in the system, and you have a IAC bus named beginning with the letters “IAC”.

## 4. Add a second instrument, to be the destination for Cthulhu

Similar to step 2, create another software instrument track, and choose an instrument (soft-synth) to be the destination for Cthulhu.

## 5. Open “Click & Ports” page in Logic's Environment window

Open Logic's Environment Window (command-8) and select the “Click and Ports” page in the top-left area of the environment window.

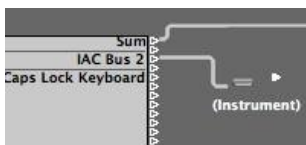


## 6. Create an 'Instrument' object in the Environment

In the menu bar along the top of Logic's Environment window, click New, and select "Instrument". You should see an object appear named "(Instrument)".

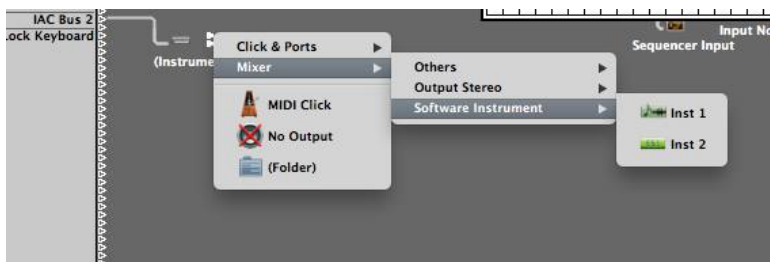
## 7. Connect the last IAC Bus to this Instrument Object

Click-drag from the triangle corresponding to your last IAC Bus in the "Physical Input" object to this instrument object we created in step 6. You should now see a wire connecting them, as pictured below:



## 8. Assign the destination of our Instrument to the synth we wish to play.

Alt-click on the triangle to the right of the newly-created instrument object. A pop-up will appear where you can select the destination track (the soft-synth you wish to have Cthulhu play).



In the illustration above, we would select "Inst 2", which is the soft-synth created as a destination for Cthulhu in step 4 of this tutorial.

## 9. Record-Enable Cthulhu in the Logic Arrange window if it is not already.

This way, incoming MIDI notes will route to Cthulhu, and Cthulhu will in-turn pass them to the IAC Bus, which in turn passes the IAC Bus MIDI to the Instrument destination (Inst 2 in this example).

## 10. Lastly, Select a Chord Preset in Cthulhu

This way incoming MIDI notes will output something!

When you play MIDI notes, you should hear Cthulhu **and** your destination soft-synth both trigger. Click the speaker icon in Cthulhu (or pull its fader down) if you only wish to hear your soft-synth.

## Option 2: MIDI Out in Logic. “Quick Method”

This saves some time if you want to get up and running with some chord generation or arp quickly. However, you need to exercise caution if following this shortcut – **DO NOT** Record-enable Cthulhu's track, or you may create a (MIDI) feedback loop which could bring your machine to its knees (Logic may hang, as MIDI coming from Cthulhu will go back to Cthulhu, ad in finitum):

Assuming you have tried the above method first (verifying IAC bus is installed):

- 1) Add Cthulhu to a track, pencil some notes in a MIDI clip
- 2) Add a Virtual Instrument e.g. ES-1
- 3) Record-Enable the ES-1 track
- 4) Load some chord preset into Cthulhu

At this point, the MIDI clip should trigger Chords in Cthulhu and you should hear Cthulhu make sound with its internal oscillator. Then the MIDI should be getting received by the ES-1 track. If you like what you hear you can press record in Logic, and your chords/arp will be committed to the ES-1 track.

# 5. Cthulhu in Logic X

---

## 1. Add a software instrument

Running Cthulhu as a MIDI FX means that Cthulhu will not be able to output actual audio (it's test tone).

So, If you don't already have an instrument in your song to use Cthulhu with, you'll need one.

In Logic X's Track menu->New Software Instrument Track.

Add an instrument to this track as you normally would.

## 2. Add a software instrument

Running Cthulhu as a MIDI FX means that Cthulhu will not be able to output actual audio (it's test tone). However, the good news is the routing (the MIDI out from Cthulhu, to the instrument on the same track) is automatic.



Cthulhu will appear in the MIDI FX menu

Add any instrument for Cthulhu to play here

Don't forget to load a Chord Preset in Cthulhu, and feed MIDI notes to the track.

# 6. Operating Cthulhu

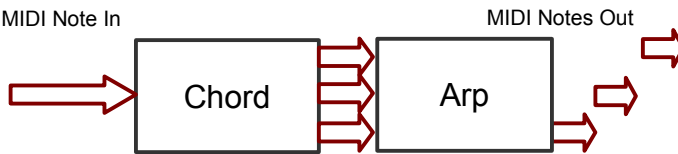
---

## Two Modules, One Plug-in

As mentioned in the introduction of the manual, Cthulhu is essentially two-plugins-in-one. There is a Chords module and a Arp Module, each with their both section of the interface, separate presets menus, etc.

**You can enable either module or both-at-once.** In the event that both modules are enabled, the Chords module is processed before the Arp.

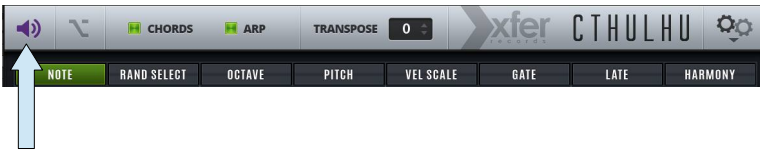
The signal flow is pictured below:



## Mute Output (MIDI only)

Cthulhu is designed to be used to trigger other software instruments, by routing MIDI from Cthulhu to the destination software instrument inside your host sequencer program, as detailed previously in the manual.

However Cthulhu will also create Sawtooth tones in order to allow you to preview chords easily. You can disable this test-tone function with the speaker icon in the very top-left corner of the plug-in. (MIDI will still output, just no audio will output, which is really the intended usage).



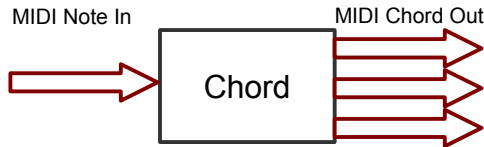
Click the purple speaker to mute the audio “test tone” output. MIDI still transmits!

# 7. Chords Module

---

## Chord – One note in becomes many.

When you send a MIDI note to Cthulhu with the Chords enabled, it sends out a MIDI chord of up to 8 notes, which is sustained until either the note is released, or a different note is received. The velocity of the notes out are based on the incoming velocity, but can be scaled between 0% and 200% for each chord-note.



There are 128 “Chord Slots” (chord memories) for a preset in Cthulhu – one for each of the 128 possible incoming MIDI note pitches. The GUI will update to indicate the last received MIDI note (chord slot) in the dark rectangle, displayed below:

**Analysis:** the chord to be output from the given chord-slot is automatically analyzed and displayed in the “thought bubble”.

**Chord-Slot:** In addition to triggering the chord, the last incoming MIDI note will also display in the dark rectangle here.

This most-recently triggered Chord-slot additionally appears visually, as a bright-orange ‘blip’ above the piano key corresponding to the last incoming MIDI note (C1).



## Used Chord Slots

Chord slots which contain a chord will be colored in **blue** just above the piano keyboard. This way you can tell visually what keys will trigger a



These chord-slots all contain a chord (blue above the piano key)

These chord-slots are empty / *do not* contain a chord (gray above the piano key)

## Altering a chord within a Chord Slot

You can see the notes which make up the chord on the piano keys in green on the piano keyboard, as well as view the note numbers themselves (e.g. F2, A2, etc. pictured above) in the middle of the plug-in, in the "1 2 3 4 5 6 7 8" voice slots.

**You may edit a note in the chord by either:**

- dragging the text-value of the note
- clicking on the red keyboard note on the piano keys (the piano key will become colored green as you drag it to a new pitch).

**You can add a note in the chord by either:**

- dragging the text-value of a voice from "OFF" to a note number
- double-clicking on a piano key

**You can remove a note in the chord by either:**

- dragging the text-value of a voice to "OFF"
- Command-click (OSX) / CTRL-click on (PC) on the red piano key



## Chord Panel Area (alphanumeric Display of Chord)



As pictured above, the chord panel area allows you to view and edit the notes that make up the currently-selected Chord-slot. The top row displays the note numbers contained in the chord.

The Second row displays the Velocity scaling for the notes in the chord (between 0% and 200% of the incoming velocity). This way you can make certain notes in the chord have more or less velocity compared to other notes, when triggered.

## Lock switches

The third row pictured above is a row of switches labeled “Lock”. These switches are not per-chord slot but rather are Global. This is a way to “lock” a certain pitch / voice of Cthulhu if you wish to have a current note-number setting hold the current value (at the time of enabling the lock) on every chord (often referred to as a 'pedal tone' in music theory).

## Macro Switches (Learn / Copy / Pasta / WTF)



### LEARN

This switch engages a “Chord Learning” mode. When you click the “Learn” LED, it will illuminate, letting you know that Cthulhu is now ready to learn the chord for any notes you play. As you play MIDI notes, you will see them get assigned (piano keys for incoming notes will turn red) for the currently selected chord slot. When you release all held MIDI Notes, the next chord slot will engage for Learn( the currently selected chord slot will advance up by one). When you are finished having Cthulhu learn chords, click the Learn switch again to turn it off.

### Learn a single chord slot

There are times when you may wish to just Learn a single chord slot, and not have Learn continue to the next chord. In this case, Command-click (OSX) / CTRL-click (PC) on the Learn switch. Play MIDI notes to Cthulhu as in regular Learn mode. Once the MIDI notes are released, Learn will shut off.

### Advanced Learn (Note On Only)

There is an Advanced Learn mode for auto-learning a bunch of chords simultaneously from a piano roll, for when a MIDI score has overlapping notes. In regular Learn mode, Cthulhu will not advance to the next chord-slot until all

notes are released. In advanced Learn, chords are learned by simultaneous note starts:

If you Alt-click on the Learn switch, it will enter the Advanced Learn mode. Since notes need to start simultaneously, this mode is intended to be used by "playing" a (quantized) MIDI file from your host sequencer in to Nerve. Regular mode could work fine for this as well, provided that there is not overlap between chords (you could shorten all durations in the MIDI score). Cthulhu looks at all simultaneous MIDI notes and creates a chord from these, then advances the currently-selected Chord Slot to the next note up.

To recap, the steps to use Advanced Learning mode are as follows:

- 1) Load a MIDI file / put the MIDI score on the track containing Cthulhu (or route the MIDI to the track containing Cthulhu).
- 2) Set the Currently Selected chord slot to the lowest note you wish to have chords on (e.g. C-1)
- 3) Alt-click on the Learn button – Cthulhu is now ready to start auto-learning chords.
- 4) (optional) If you are impatient, you may set your sequencer tempo very fast, Cthulhu is a fast learner :)
- 5) Press play in your Host to have the MIDI file play in to Cthulhu. You should notice the red-blip advance along at each new midi note-chord, and the slots to the left filling becoming blue.
- 6) Click on the LERN button so the LED is dark, to deactivate the Learning mode.

## Copy Button

The Copy switch is used for Copying the currently-selected Chord-slot to memory, to be Pasted elsewhere with the Pasta switch.

## Pasta Button

This will store the Copied chordslot to the currently selected Chord-slot.

## WTF Button

This button will create a random chord on the currently-selected Chord-slot. You can use the locks in a temporary manner, to prevent certain notes from changing when you use this feature..

# 8. Chord Menu

---



In the chord section of Cthulhu Cthulhu is a chord alteration menu. It contains functions **which affect ALL chord-slots**.

The menu is accessed by clicking the “Menu” button pictured above. This brings up some options:



## “Sort” Sub-Menu

Selecting an item in the Sort sub-menu will re-sort (re-order) the locations of the Chord-slots. This is useful for giving yourself a new playing-perspective on the chord-slots in memory:

- **Sort By Low (Absolute)** - This re-orders the Chord-slot locations, based on the lowest note number present in every chord.
- **Sort By Low (Pitch Class)** - This is the same as above, except that instead of sorting by the actual note number, it sorts by the note-name of the lowest note (e.g. all chords with a C as the lowest note are grouped together, irregardless of what octave the C is located in).
- **Sort By High Note** - This re-orders the Chord-slots to be sorted by the highest (absolute) note number in each chord.
- **Sort By Number of Notes** - The Chord-slots are re-ordered based on how many notes are present in each chord (e.g. 5-note chords are grouped together, followed by all 4-note chords grouped together, etc).
- **Sort By Hindemith (Chromatic)** - This attempts to analyze each chord, to determine the root key, based on Hindemith Analysis (lowest note of the first perfect fifth is the root, if none is present the highest note of a perfect fourth is the root, etc). The Chord-Slots are then grouped together by the analysed root (so all chords that make up a C chord are grouped, followed by all chords that make up a C# chord, etc).
- **Sort By Hindemith (Circle of Fifths)** – Similar to the above but the resulting chords are grouped in a circle-of-fifths order (C chords, then G chords, then D chords, etc) instead of chromatic (C chords, then C# chords, etc).

## “Remove” Sub-Menu

The Remove Sub-Menu allows you to erase chord-slots you may not want, based on a few different options. Keep in mind after using a Remove function,

---

the chord locations will move (as the used chords get collapsed together).

- **Remove Duplicates** – All chords are compared to look for any identical chords. The duplicates are removed.
- **Remove One-Note Chords** – Any Chord-slots consisting of a single MIDI note / voice will be removed.
- **Remove Two-Note Chords** – Any Chord-slots consisting of two MIDI notes / voices will be removed.
- **Remove Three-Note Chords** – Any Chord-slots consisting of three MIDI notes / voices will be removed.

## Make Major Chords Minor

This will scan all Chord-slots for Major chords (based on Hindemith Analysis) and will alter the third-scale-degree of the chord if major (+4 semitones above root) it becomes minor (+3 semitones above root).

## Make Minor Chords Major

Similarly, This will scan all Chord-slots for Minor chords (based on Hindemith Analysis) and will alter the third-scale-degree of the chord if minor (+3 semitones above root) it becomes major (+4 semitones above root).

## Make All Chords Suspended

This will scan all Chord-slots for Minor and Major chords (based on Hindemith Analysis) and will alter the third-scale-degree of the chord if minor or major (+3 or +4 semitones above root) it becomes a suspended fourth (+5 semitones above root).

## Save Preset (Chords+Arp)

This particular selection (Chords+Arp) saves the entire state of Cthulhu. This data automatically saves inside your host song, so there is no need to use this feature unless you want to have a starting point to use in other projects. This may be useful in some situations.

This menu item will bring up a dialog window prompting to choose a name (and optionally a new location) for the preset.

## Save Chords Preset

This menu item will only save the chord info (lower-half of Cthulhu GUI). This is useful if you wish to have your chords appear in the Chords presets menu, for future use.

This menu item will bring up a dialog window prompting to choose a name (and optionally a new location) for the preset. By default they will save in the “Cthulhu Presets/Chords/” folder, as this is the folder scanned for the Chord Presets menu.

## Save Arp Preset

This menu item will only save the Arp info (upper-half of Cthulhu GUI), all 12 patterns and corresponding controls (e.g. Swing and Gate controls). This is useful if you wish to have your current Arpeggiator state (patterns, etc). appear in the Arp presets menu, for future use.

This menu item will bring up a dialog window prompting to choose a name (and optionally a new location) for the preset. By default they will save in the “Cthulhu Presets/Arp/” folder, as this is the folder scanned for the Arp Presets menu.

## Import MIDI File

This menu item allows you to import chords into Cthulhu from a standard MIDI file. The chords import beginning with the currently-selected chord slot. When up a dialog for a single-track MIDI File (type 0 MIDI File). Type 0 Midi files have a single track of data. Once loaded, Cthulhu will determine where a chord begins and ends based on held notes – if notes start or end irregularly (not all simultaneously), Cthulhu assumes all notes are a chord as long as a note is held – in other words, where there are no notes playing in the file, Cthulhu will advance to the next chord-slot.

If MIDI File does not import:

As mentioned above, The MIDI needs to be in a type 0 (single-track) MIDI file. Many downloaded MIDI files are in type 1 (multi-track). You can either import the file in to your host program and manually merge (copy/paste) the relevant parts into a single track and export this, or you could also try a utility which automates the type1 to type0 process, such as MIDI Squeezer:  
<http://www.cellspark.com/MidiSqueezer.html>

All considered, it is recommended to edit 'random' MIDI files into chords, as many pieces contain a lot of melodic notes, etc. which won't necessarily make the best source material for Cthulhu.

# 9. Arpeggiator

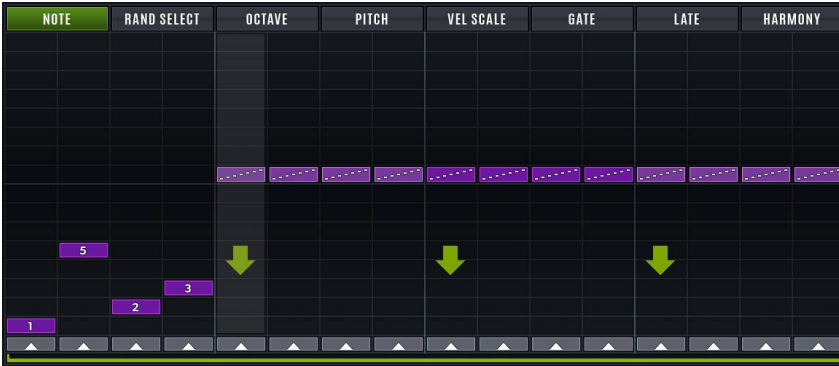
---

## Arpeggiator – many notes in output one-at-a-time

The Arpeggiator is a tempo-synced effect. This means your host transport should not be stopped, so in order to hear the arpeggiator, you should have your host playing.

### Graphs

Unlike most arpeggiators, Cthulhu has a graph area (pictured below):

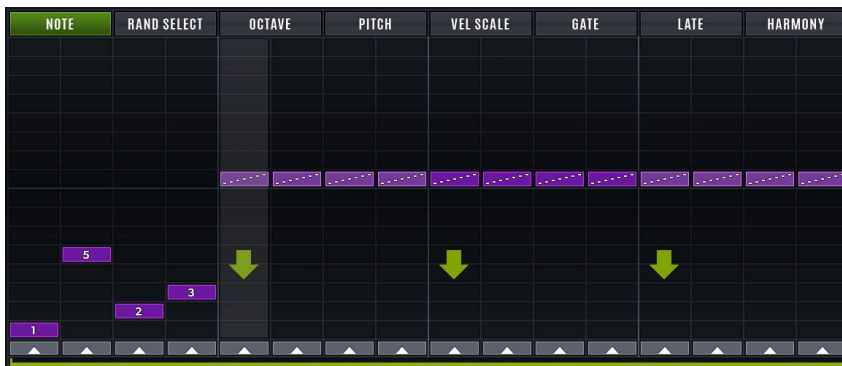


Notice the tabs along the top. These are referred to below as Graph Tabs, as they determine which data content of the pattern you are editing.

# 10.Arpeggiator Graphs

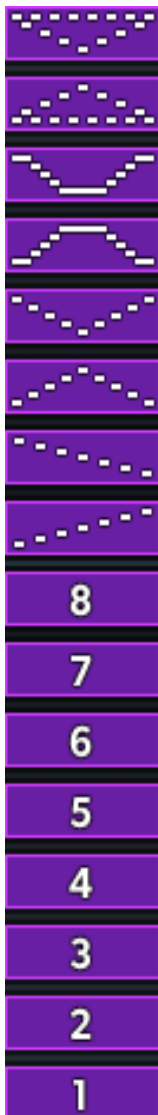
---

## Note tab



The Note graph (pictured above) allows you to choose what happens on a given step for the arpeggiator. If you click on any step (left-to-right) and drag up/down, you will see a different symbol on the step. The top-half of the graph is various arpeggiator patterns, which act like a typical arpeggiator, where the note output varies consecutively one step after another, e.g. if you have a triad held down, the output will be 1-3-5-1-3-5-1-3-5 etc.

These markings are shown below with description, from bottom-to-top on the step:



**fingered top** - every 2<sup>nd</sup> note is the high note of the chord.

**fingered bottom** - every 2<sup>nd</sup> note is the low note of the chord.

**down and up** - all notes down then up, with both extremes (highest and lowest note) sounding twice.

**up and down** all notes up then down, with both extremes (highest and lowest notes) sounding twice.

**down/up**, similar to down and up, but the highest and lowest notes only sound once.

**up/down**, similar to down and up, but the highest and lowest notes only sound once.

**down** - notes are played from highest to lowest pitch on consecutive steps.

**up** - notes are played from lowest to highest pitch on consecutive steps.

8

8 - The eighth-highest note of the chord plays on this step.

7

7 - The seventh-highest note of the chord plays on this step.

6

6 - The sixth-highest note of the chord plays on this step.

5

5 - The fifth-highest note of the chord plays on this step.

4

4 - The fourth-highest note of the chord plays on this step.

3

3 - The third-highest note of the chord plays on this step.

2

2 - The second-highest note of the chord plays on this step.

1

1 - The lowest note of the chord plays on this step.

## Step muting

If you click on a step and drag all the way down below "1", you will see





applied.

Above halfway means that the output step will be greater than the assigned step, e.g. if the Note Sel step is set to 2, and the random value is set to 2 above middle, then every time this step occurs, the Arp will output “2”, “3” or “4” (the 2<sup>nd</sup>, 3<sup>rd</sup>, or 4<sup>th</sup> note of the incoming chord).

## Octave tab

The third graph-selector along the top of the plugin says “Octave”. Selecting this will display the Octave pattern. This allows you to have the output notes transposed +/- 8 octaves from the otherwise calculated output note. The default value is along the middle, which means no octave transposition is applied.

## Pitch tab

The pitch graph-selector will allow notes to get transposed by semitones. However this has a special function which may take a minute (or hour) to comprehend (depending on your music theory background).

Simply put, transposition is only applied to chosen scale degrees. This makes it possible to have a chord change flavor (e.g. minor to major, if you only enable '3' on the Pitch Enable buttons), without the root note or fifth of the chord ever transposing.

Ctulhu analyzes incoming chords and determines what the root of the chord is, even if the chord is played in an inversion. For instance, a triad G-C-E is determined to be a C (Major) chord, as is C-E-G, as is E-G-C. This root information is used on the pitch graph.

See the following image:



Pitch Enable buttons

*When the Pitch graph is selected* (top of image, in green), you will see numbers appear on the pitch enable buttons (bottom of image). **You must click on these numbers so they turn green (enable), in order to choose which scale degrees of the incoming chord are eligible for pitch transposition.** If you enable all 7 of these buttons, then the pitch graph will transpose any/all notes of the chord.

## **Vel scale tab**

The Vel scale tab allows the velocity to be scaled on a per-step basis, which can create interesting variation if your destination instrument is set to respond to velocity (e.g. Velocity mapped to filter cutoff).

## **Gate tab**

The Gate tab will shorten the duration of steps. The default is 100% (full vertical blue bar, no duration change). This is useful for making notes shorter on certain steps, to create an additional level of rhythmic motion to the output sound.

## **Late tab**

The Late tab repositions the timing of the step getting triggered to be early or late. This enables a custom-swing feel to the Arp, for instance to have a certain step intentionally sound early or late.

## **Harmony tab**

The Harmony tab adds a second note to be output for particular steps, this is useful if your destination synth is polyphonic (can handle making multiple notes play at once). The note is relative to the output pitch for this step, between an octave down and an octave up.

# 11. Additional Arp Features

---

## Arp Patterns



The arp has 12 separate patterns in memory at once. The pattern selector allows you to view which pattern is selected/active, as well as which patterns are not in the default state. For example in the picture above, pattern B is selected/active (as it is highlighted green), and patterns A thru D contain pattern data (as they are green, not grey).

Clicking on the letters A-L will select a new current pattern to view and use.

The other controls pictured above are also pattern-specific settings, in other words, they can have a separate value/setting for each pattern, and are described below:

**Rate knob:** this knob controls the length of a step, from 16 bars (very slow) to very fast (1/512<sup>th</sup> note). Arp settings are most commonly somewhere in the middle (8<sup>th</sup> to 64<sup>th</sup> notes, with 16<sup>th</sup> notes being the most common value, hence it is the default).

**Retrigger switch:** When enabled, the Arp will retrigger when a MIDI note is received, in other words, playback will begin again at step1 at this very moment.

**Link Lengths switch:** This switch, when enabled, will link the Graph Pattern Length bar (mentioned earlier, see “Graph Pattern Length” in the previous chapter) so that it will change for all graph-pages. This way you don’t need to set the length on all 8 graph-pages if you want them all the same.

**Chord Mode:** When chord mode is enabled, the Arp will additionally output all of the notes which are incoming, transposed upwards so the intended arp output note is the lowest heard. This is useful for making “chord arpeggios”, similar to playing the inversions of a triad on a piano (e.g. C-E-G E-G-C G-C-E).

**Pattern Name:** You can click here and type a name for each pattern.

## Note output prevention



Because the pitch graph can transpose individual step-pitches, in some situations you could create notes which are not in the key of your song, and therefore potentially undesirable. In order to help mitigate these situations, you can prevent certain notes from getting output.

There is a single-octave of notes (pictured above) in the Arp section. In addition to displaying the current output note(s), you can click on these piano keys to block certain notes from being output. There are 4 separate states for the switch, which are cycled through on each click:

- black/white (default) – note is output as expected.
- red (solid red) – this note is blocked. The note will output (silence).
- red (with > symbol) – this note is blocked, a higher pitch will sound instead (plus one semitone).
- red (with < symbol) – this note is blocked, a lower pitch will sound instead (minus one semitone).



## Clock Div

The number box to the left of the Clock Div slows down the rate of the currently-selected graph. This number represents a denominator of the playback speed for the visible graph (e.g. "2" =  $\frac{1}{2}$  speed, "4" =  $\frac{1}{4}$  speed, etc).

## Copy/Pasta/Rand



Three buttons to the left of the Clock Div are the Arp's Copy, Paste, and Random buttons, respectively. These function on the currently-selected graph only. If you wish to copy an entire pattern (all graphs), simply alt-drag the pattern button/letter (e.g. 'A') to the desired destination (e.g. 'B').

# 12.Presets

---

## Presets Browser / Loading Presets



The **Chord** Presets Browser appears in the top-right corner of the Chord Section. Similarly, the **Arp** Presets Browser appears in the lower-left corner of the Arp section. The text displays the currently loaded preset. Clicking on the text opens a pop-up menu with which to load any of the presets. The arrow buttons to the right of the menus are “shortcuts” to advance you through the presets in the last-selected subfolder in the menu. The arrows only appear once a preset is selected.

## Anatomy of a Chord preset

A Chord preset contains the entire 'state' of Chords module in Cthulhu:

- 128 Chords (each with 8 chord notes + velocity scalings)
- Global Note-Out Transpose (accessible as a VST Parameter)
- Last Selected Chord (displayed in top-left of chords module, and as bright orange 'blip' on chords keyboard)
- “Lock” switches for the 8 chord voice-slots.

## Anatomy of an Arp preset

An Arp preset contains the entire 'state' of Arp module in Cthulhu:

- Currently-selected Pattern (A thru L)
- Names of the patterns (A thru L)
- Pattern Data (step sequencer step values/lengths/rate/etc)
- Note constraints (single-octave piano display)

All preset types are saved with the .fxp (VST Preset) file extension. However there are three different formats, for each preset module type. Thus, It is strongly recommended to save the presets to the default corresponding folders, which automatically appear when saving a preset:

Chord Presets:	/Cthulhu Presets/Chord/
Arp Presets:	/Cthulhu Presets/Arp/
Chord+Arp Presets:	/Cthulhu Presets/Presets/

This way your presets will appear in the appropriate preset menu.

# 13. MIDI CC and Addendum

---

This chapter documents the changes since the 1.0 release.

## Version 1.1:

- new GUI graphics (redesigned by Lance Thackeray)
- GUI now can be resized to 200% (click in top-right corner where 'gears' display)
- Arp Free Rate (Hz) option (same menu in top-right) which puts the Arp clock to a free time instead of tempo-synced
- add prefs in CthuluCon fig.txt for
  1. Speaker/Test-Tone Enabled by default (1 = on)
  2. default GUI size (0 = 100%, 1 = 200%)
- Chords module now defaults to minor triads on all chord slots instead of silence on all chord slots (however you can select “-init-” from the Chord Presets menu if you want silence)
- MIDI CCs (on Ch1) now pass through to the destination instrument

## version 1.01 through 1.09:

- added: Transform sub-menu on Chords Menu. Now with 2 additional choices:
  - By Arp Note-Prevent Rules: looks at the single-octave on the arp of note prevents (and transpose directions) and applies these rules to all chords
  - By Selected Chord Notes Only: similar to the above, but transposes all chord notes on all chord slots to match the notes of the selected chord slot.
- - added: drag-and-drop Cthulu monster to host for currently selected chord as a MIDI File
- - added: alt-drag-and-drop A-L button to host for 1-bar render of Arp pattern
- - added Song Options submenu: midi out separations for chord notes, optionally send notes to different channels 1 to 8, or CH.1 for full chord, 2 for arpeggio, 3 to 10 for notes (this menu is not accessible in Ableton Live, since multi-MIDIOut is not supported by Ableton currently).
- added: "Watch" (eye switch) which works in conjunction with Menu->Remove->Unwatched Chords: Click the eye switch, play the chords you want to keep, then select Menu->Remove->Unwatched chords and all chords which were not triggered will be erased.  
added: Menu->Transform->Populate Empty Slots with Variations: creates transposed "inversions" for all loaded chord slots.  
added: shift-click learn switch = learn with timer: chord will not be learned



until the chord is held down for 1 full second. This allows you to try out chords quickly without actually them.

added: "remove duplicates" now re-sorts the notes of all chords low-to-high, to catch duplicates which may have had a different note ordering.

added: shift-click on piano to trigger chord; alt-shift-click to do both functions (change to a different chord slot and trigger it)

- beta 9:  
added: typeable values for notes/velocities (double-click on number boxes)  
added: import of .chords files from piz MIDIChords
- beta 11b:  
added: MIDI latching option (switch at top upper-left). When enabled, incoming MIDI note-off's are "ignored" / incoming notes will hold until a new note/chord is sent to Cthulhu.  
added: pass through aftertouch / pitch bend / program changes to downstream MIDI devices  
fixed: import MIDI now more robust (compatible with Logic's Type0 SMF export 'abnormalities').
- beta 14h:  
the following feature is VST Only, as Logic does not play nice (sends CC7 on all MIDI channels, when you press play or stop).  
If anyone sees steps unexpectedly changing in a VST host (most likely steps 1 or 7 of the note selection graph) please let me know, I have a feeling I'll need to add some sort of "enable" for this feature.

CC's on Ch2 or Ch3 control the graphs in a realtime fashion. I may have to make it an option you enable in the config because I believe some hosts like to send zero-midwheel messages, and god knows what else, at their own choosing.

I

e.g.

CC1 CH2 = 'Note Sel' graph (current step)

CC2 CH2 = 'Rand Sel' graph (current step)

CC3 CH2 = 'Octave' graph (current step)

CC4 CH2 = 'Pitch' graph (current step)

CC5 CH2 = 'Gate' graph (current step)

CC6 CH2 = 'Late' graph (current step)

CC7 CH2 = 'harmony' graph (current step)

and then CH3 can access every step, e.g.

CC1 CH3 = 'Note Sel' graph, step 1

CC2 CH3 = 'Note Sel' graph, step 2

CC17 CH3 = 'Rand Sel' graph, step 1

CC49 CH3 = 'Pitch' graph, step 1

[VST] CC on MIDI ch 2 and 3 are disabled by default, so your arp pattern won't mysteriously 'blank' steps out. If you have used or want the ability (added in b14h, see below in the change log), this feature can be enabled with CthulhuConfig.txt found in the top level of the Cthulhu Presets folder.

1.0.6 - MIDI CC's to control current chordslot note assignments (CC101-108) are disabled by default as they were causing some support/grievances. If you want this feature you can enable it in CthulhuConfig.txt which is atop your Cthulhu Presets folder (see above for location of this folder).

