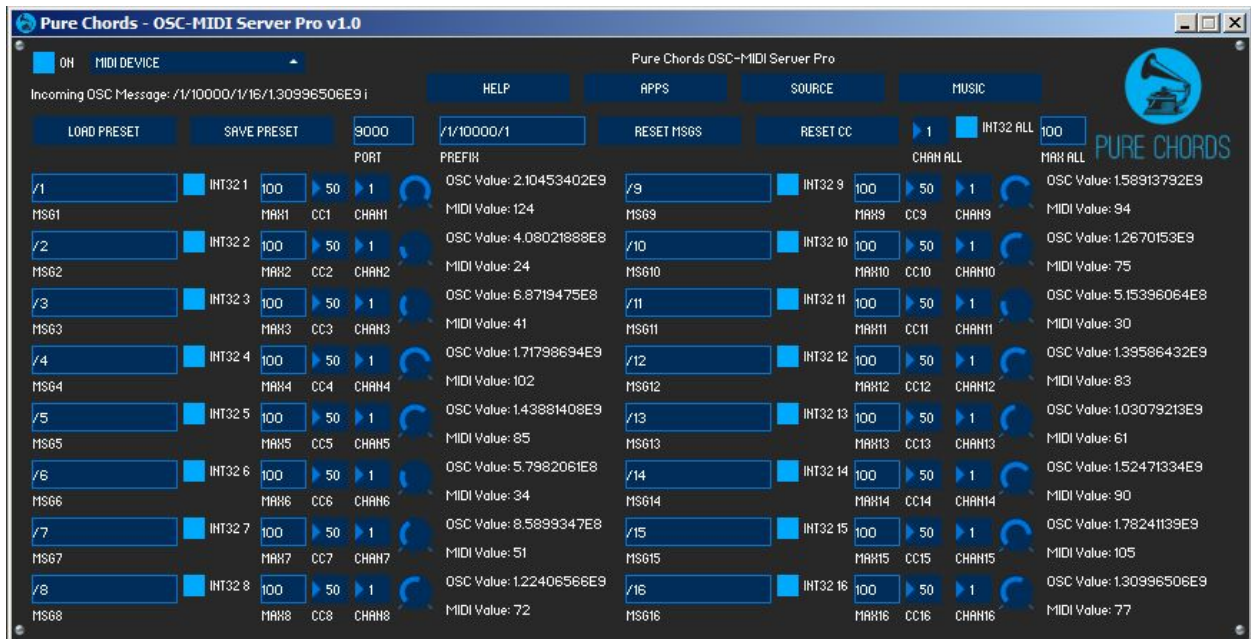


Pure Chords OSC-MIDI Server Pro

V1.0



Pure Chords OSC-MIDI Server Pro lets you easily transform Open Sound Protocol (OSC) messages into MIDI. Control vintage MIDI hardware, VST plugins, DAWs and any other audio application with simple MIDI control change (CC) values. Sixteen instances each listening on a unique OSC address and transmitting values to a unique MIDI channel and CC number. Both incoming OSC and outgoing MIDI is monitored providing quick and intuitive configuration between addresses, ranges and values.

Make sure to set up your system to work with Open Sound Protocol networking! Route the ports you would like the server to listen on to your machine and let the server pass through your firewall.

Network and Firewall setup example and guide on Windows 7:

<http://faxinadu.net/purechords/app/Network%20Setup%20For%20OSC%20Control.pdf>

Controls

Links

Help: Launches a browser with a link to this manual.

Apps: Launches a browser with Pure Chords on the Google Play Store where you can grab an assortment of OSC capable apps for Android to use with this server.

Source: Launches a browser with a GitHub sources code repository for this server.

Music: Launches a browser with Faxi Nadu and Ocean Star Empire on Bandcamp.

Settings

On: Turn the server on and off. Sets and releases the resources - the selected MIDI driver and the selected network ports.

MIDI Device: Selects the MIDI driver the server will output MIDI to. If you intend to use the server within your workflow together with a DAW or other musical program (send MIDI for the server to the DAW and control VST plugins for example) then you must have available MIDI drivers for both your DAW and the server. If that is not available with your setup look into a virtual midi driver like loopMIDI, LoopBe1 or Hubi's Loopback MIDI Driver (or any other) to add some virtual drivers to your system.

Incoming OSC message: A monitor for the complete raw incoming OSC messages.

Load Preset: Select and load a preset file.

Save Preset: Save the current server settings to a preset file.

Port: Sets the local port on which the server will listen on for incoming OSC messages.

Global Controls

Message Prefix: Together with each instance's unique message, the prefix sets the unique OSC message address listen's for (see instance message field below).

Reset Messages: Reset all the instance messages to default values at once. (/1-/16).

Reset CC: Reset all the MIDI CC values to default (102-117)

Chan All: Set the MIDI channels of all instances at once.

Mode All: Set Int32 mode on or off for all instances at once.

Max All: Set the maximum possible incoming value for all instances at once.

Instance controls

Message: Set the unique OSC message address each instance will listen for. This is appended to the message prefix to form the unique OSC message address. For example if /1/1 is set in the prefix field and /1 is set in the unique instance message field then the instance will be listening for messages with an address of /1/1/1.

Mode: Set Int32 mode on or off for each instance. This is useful when an OSC program is sending the server values in the int32 range. Setting the mode to Int32 disables the max field of the instance and instead uses int32 as the maximum possible incoming value. In practice this scales the incoming int32 OSC message to MIDI values.

Max: Set the maximum possible incoming value for each instance. The server will scale your incoming values accordingly into MIDI values. For example if your app is sending the server a knob that has a full range between 0-100 then set the max field to 100. For another example if your app is sending a 0-1 floating point value then set your max field to 1. When the instance is in Int32 mode, the max field is disabled.

CC: Set the MIDI CC number each instance will output values to.

Chan: Set the MIDI channel number each instance will output values to.

Value Monitors: Incoming OSC message and outputting MIDI value monitors for each instance.



PURE CHORDS

<https://www.facebook.com/purechordsrecords>

<http://www.faxinadu.net>

Developed by Yaron Eshkar. Written in Java using the Processing 3.3.7 IDE

The development of this software was made possible using the following library components:

[themidibus](#) by sparks

Licensed Under: [LGPL-3.0](#)

[oscp5](#) by sojamo

Licensed Under: [LGPL-2.1](#)

[controlp5](#) by sojamo

Licensed Under: [LGPL-2.1](#)