

## Bitwig CLAP

### A New Audio Plug-in Standard

Bitwig and u-he announce CLAP (“CLever Audio Plug-in API”), the new open standard for audio plug-ins and hosts. CLAP offers modern features, innate stability, and rapid support for plug-in and host developers. And since it's open source and liberally licensed, CLAP is a safe bet for the future. Developed in collaboration with experts from diverse fields in the music software industry, CLAP is a cutting-edge plug-in standard, designed for modern computers, software, and paradigms. CLAP caters to novel DAW concepts, and opens up new horizons for what a plug-in can do or be.

Developed with modern CPUs in mind, CLAP takes multi-thread management to a new level, with a clear and efficient allocation of roles between plug-in and host. Specifically, CLAP allows collaborative multicore support between plug-in and host through a so-called “thread-pool”, also allowing hosts to manage CPU-threading for plug-ins that provide their own multicore support. Preliminary tests show significant performance gains compared with current solutions.

CLAP hosts can read plug-in metadata and help organize your plug-ins. As CLAP hosts can retrieve information from plug-ins without having to wait for them to initialize, plug-in scans can be much faster. Furthermore, we’re currently finalizing an extension which lets plug-ins tell the host which files they need (e.g. samples or wavetables), and the host can consolidate those in the project file. That means you'll never lose a sample while transferring a project between systems!

The CLAP standard promotes new ways to create music with automation, modulation, and expressions.

- CLAP supports per-note automation and modulation (in accordance with the recent MIDI 2.0 specifications).
- CLAP’s parameter modulation concept also allows for temporary parameter offsets. Parameter modulation is non-destructive, so as soon as the modulation has finished, the target parameter will return to its original state.
- CLAP makes it possible for polyphonic plug-ins to have their per-voice parameters modulated for individual notes (“MPE on steroids”).

With this new standard we aim to inspire host developers to add exciting new features to their products. Initial implementations by Bitwig, u-he and the Surge project demonstrate just a few of the possibilities.

The KISS (keep it simple, silly) mantra belongs to CLAP. From the C-only ABI, which allows binding to any programming language, to the transparent client-server model between host and plug-in, the robustness and clarity of the threading model,

and the single event queue for all kinds of parameter changes, timing and MIDI. Despite being so comprehensive, everything in CLAP is easy to find and easy to implement.

A single cpp/hpp glue layer for C++ offers a quick start into the ABI, and its built-in “proxy layer” finds common threading bugs in an instant. Apropos C-only ABI: There are no platform specific dependencies: In theory CLAP should also run well on embedded platforms, game consoles or specialized supercomputers.

CLAP is open source, released under the MIT license: No fees, memberships or proprietary license agreements are required before developing or distributing a CLAP capable host or plug-in, and the license never expires. In a nutshell, there are no entry hurdles for developers, and plenty of open source projects already benefit from CLAP.

Inspired by MPE and MIDI 2.0, CLAP can adapt to any future MIDI standard. Wherever a solid standard exists, we allow developers to freely decide how to apply it. Plug-ins can receive and send any MIDI event on multiple ports.

A simple system for extensions makes CLAP future-proof. Companies can offer proprietary extensions if required for specific features. CLAP 1.0 is the result of a multi-year project initiated by u-he and Bitwig, with design and implementation contributions by a group of commercial and open source audio developers from across our industry. As we proceed beyond the initial set of extensions, we are committed to establishing a transparent process to govern the standard which allows participation from the entire audio software community. We welcome participation from the development community today, and we will share details of these processes and governing models over the second half of 2022.

CLAP does not only exist on paper; there are several ways you can try it out now. See the links below for projects from u-he, Bitwig, and others. Several other industry players are currently evaluating the CLAP standard.

CLAP is now available in the latest Bitwig Studio beta version and in the u-he CLAP beta versions of ACE, Diva, Hive 2 and MFM2.5. It has also been implemented by Chowdhury DSP, MultitrackStudio and the Surge Synth Team.

### Open Source Projects Using CLAP:

- Avendish project
- CLAP JUCE Extensions project
- Dexed
- MIP2 framework
- nih-plug framework
- TheWaveWarden (Odin2)

Following companies and projects are already evaluating CLAP for their host and

plug-in software:

- Arturia
- Avid
- BespokeSynth
- CableGuys
- Cockos
- Cytomic
- DMGAudio
- Epic Games (Unreal Engine)
- Expressive E
- FabFilter
- Image-Line
- iPlug2 framework
- LHI Audio
- Node Audio (Etonal Studio)
- Oddsound
- Presonus
- Plogue
- Qtractor
- Togu Audioline
- Valhalla DSP
- VCV
- Vital Audio
- Xfer Records

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

[www.u-he.com](http://www.u-he.com)