dylanmtaylor docs: drop references to comp...	31f0389 · 3 hours ago	
.cargo	Foundation: time, aud...	2 weeks ago
.github/workflows	ci: add nix check work...	3 days ago
apps	Merge main into #319	5 days ago
assets/app-icon	Release pipeline: sign...	2 weeks ago
contributors	Fall back to a system ...	5 days ago
crates	ProRes: check frame d...	2 days ago
docs	Merge main into #413	5 days ago
packaging	macOS: build the DM...	last week
xtask	UI.9: reconcile upstrea...	last week
.gitignore	fix(Play Audio): Resum...	last week
AGENTS.md	Release 0.5.0: linux-ris...	last week
ATTRIBUTION.md	macOS DMG: branded...	last week
CLAUDE.md	Foundation: time, aud...	2 weeks ago
Cargo.lock	Release 0.9.0	16 hours ago
Cargo.toml	Release 0.9.0	16 hours ago
LICENSE-APACHE	Foundation: time, aud...	2 weeks ago
LICENSE-MIT	Foundation: time, aud...	2 weeks ago
NOTICE	Foundation: time, aud...	2 weeks ago
README.md	docs: drop references ...	3 hours ago
ROADMAP.md	Merge main into #410...	5 days ago
clippy.toml	Foundation: time, aud...	2 weeks ago
rustfmt.toml	Foundation: time, aud...	2 weeks ago

About

An open-source, clean-room reimplementation of Avid Pro Tools in pure Rust

getartcraft.com/apps/soun...

Readme · Apache-2.0, MIT licenses found · Activity · Custom properties · 3.2k stars · 36 watching · 2.0k forks · Report repository

Releases 6

SoundCraft v0.8.0 Latest · 18 hours ago · + 5 releases

Contributors 31

+ 17 contributors

Languages

Rust 98.9% · Other 1.1%

README · Apache-2.0 license · MIT license

ARTCRAFT

SoundCraft

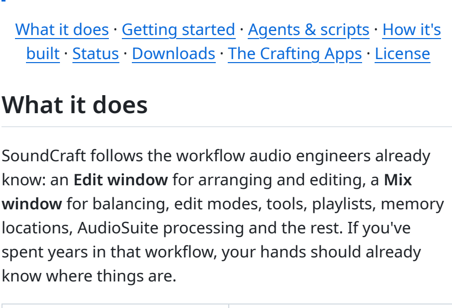
Recording, editing and mixing audio; an open-source, modern digital audio workstation (DAW), built in pure Rust.

A complete digital audio workstation that runs natively on macOS, Windows, Linux and FreeBSD, and in the browser. Multitrack editing, a full mixer with plugins, sends and automation, MIDI, recording and bouncing, and every action scriptable from the command line or by an AI agent.

100% Rust · UI egui · macOS · Windows · Linux · FreeBSD · Web · license MIT OR Apache-2.0 · status pre-alpha

JOIN US ON DISCORD

SoundCraft on getartcraft.com · ArtCraft · All Crafting Apps



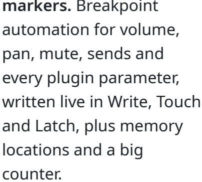
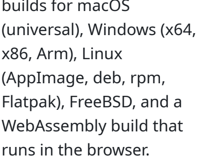
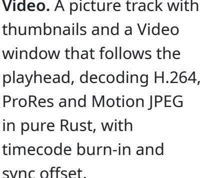
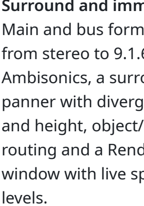
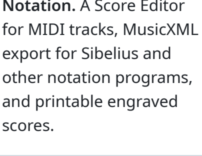
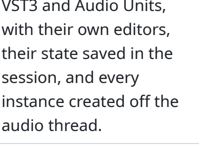
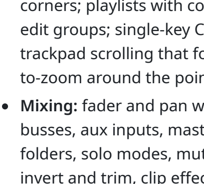
The Edit window: tracks, waveforms, MIDI, rulers, markers and the transport, playing the built-in demo song (every sound in it is synthesised by SoundCraft itself).

> Note
>
> **ArtCraft is a community of artists from all walks of life.** Painters, photographers, filmmakers, illustrators, designers, animators, hobbyists, and people who picked up a pencil last week. If you make things, you're one of us. Come say hi on Discord.

What it does · Getting started · Agents & scripts · How it's built · Status · Downloads · The Crafting Apps · License

What it does

SoundCraft follows the workflow audio engineers already know: an **Edit window** for arranging and editing, a **Mix window** for balancing, edit modes, tools, playlists, memory locations, AudioSuite processing and the rest. If you've spent years in that workflow, your hands should already know where things are.

	
Mix window. Ten inserts and ten sends per track, busses and aux inputs, master faders, VCAs, routing folders, solo-in-place with implicit solo, and meters with peak hold.	**MIDI editor.** Instrument tracks with built-in synths, a piano roll with a velocity lane, step input, quantize, transpose, and Standard MIDI File import and export.
	
Automation and markers. Breakpoint automation for volume, pan, mute, sends and every plugin parameter, written live in Write, Touch and Latch, plus memory locations and a big counter.	**Everywhere.** Signed builds for macOS (universal), Windows (x64, x86, Arm), Linux (AppImage, deb, rpm, Flatpak), FreeBSD, and a WebAssembly build that runs in the browser.
	
Video. A picture track with thumbnails and a Video window that follows the playhead, decoding H.264, ProRes and Motion JPEG in pure Rust, with timecode burn-in and sync offset.	**Surround and immersive.** Main and bus formats from stereo to 9.1.6 and Ambisonics, a surround panner with divergence and height, object/bed routing and a Renderer window with live speaker levels.
	
Notation. A Score Editor for MIDI tracks, MusicXML export for Sibelius and other notation programs, and printable engraved scores.	**Your plugins too.** CLAP, VST3 and Audio Units, with their own editors, their state saved in the session, and every instance created off the audio thread.

Feature tour

- **Editing:** Shuffle, Slip, Spot and Grid modes; Smart, Selector, Grabber, Trim, Pencil, Zoom and Scrubber tools; cut, copy, paste, duplicate, repeat, shift, insert silence, separate (at selection, on grid, at transients), heal, consolidate, strip silence, nudge; fades and crossfades in five shapes, drawn straight from the clip corners; playlists with comping lanes; clip gain lines; edit groups; single-key Commands Focus shortcuts; trackpad scrolling that follows your fingers and pinch-to-zoom around the pointer; undo for everything.
- **Mixing:** fader and pan with automation, pre/post sends, busses, aux inputs, master faders, VCA masters, routing folders, solo modes, mute and solo groups, phase invert and trim, clip effects, and automatic plugin delay compensation.
- **Automation:** volume, pan, mute, send and every plugin parameter; Write, Touch, Latch and Trim passes recorded live from the faders; thin, glide, coalesce and convert to clip gain.
- **Plugins:** 7-band and 1-band EQ, compressor/limiter, expander/gate, de-esser, maximizer, channel strip, room and plate reverbs, mod delay, chorus, flanger, phaser, saturator, lo-fi, rectifier, pitch shifter, time shift, gain, trim, invert, DC removal, signal generator, dither, a subtractive synth, a drum synth and a sampler (plays any WAV, AIFF, FLAC, MP3 or OGG across the keyboard), all original. **CLAP plugins** load too. Any processor runs offline as AudioSuite, and user presets save per plugin.
- **MIDI:** instrument and MIDI tracks, a piano-roll MIDI editor with a velocity lane, an event list, a score view, step input, quantize, transpose, real-time properties, Standard MIDI Files, and Audio-to-MIDI from a sung or played melody.
- **Recording:** punch in on a selection or on the fly, pre/post-roll, loop recording with a playlist per take, input monitoring through the full channel strip, and autosave with recovery.
- **Time and tempo:** Bars|Beats, Min:Secs, Timecode (every rate, drop-frame included), Feet+Frames and Samples; tempo and meter maps with a tempo editor; linear, parabolic and S-curve tempo ramps; key signatures and detected chords; Beat Detective and Identify Beat; pitch-preserving Elastic warping.
- **Audio files:** WAV/BWF/RF64, AIFF/AIFC and FLAC in and out; MP3, Ogg Vorbis, AAC, ALAC, CAF and the audio of MP4/MOV movies in. Any sample rate and bit depth.
- **Bounce:** mix or stems to WAV, AIFF or FLAC at 16/24/32-bit float with dither and normalisation, reporting peak, true peak and integrated loudness (LUFS).
- **Fast:** a parallel mix engine, copy-on-write undo, and a UI that holds 120 frames per second with seventy-odd tracks playing.

Getting started

Download a build from the releases page, or build from source with Rust 1.95 or later for the desktop/web UI (the core workspace's declared minimum remains Rust 1.90):

```
git clone https://github.com/storytold/soundcraft
cd soundcraft
cargo run --release -p soundcraft -- --demo
```

Linux needs the ALSA headers (libasound2-dev on Debian/Ubuntu, alsa-lib-devel on Fedora).

The web build:

```
cd apps/soundcraft-web && trunk serve --release
```

Useful shortcuts: Space play/stop, ⌘ = Mix/Edit, F1 – F4 edit modes, F5 – F10 tools, ⌘ E separate, ⌘ D duplicate, Enter new marker, ⌘ ⇧ N new tracks. **Setup › Keyboard Shortcuts** lists them all.

Logs

The desktop app writes its log records to standard error and to logs/soundcraft.log in the settings directory, beside ui.json , Autosave/ and Presets/ (Linux ~/.config/soundcraft/logs/ , or $XDG_CONFIG_HOME/soundcraft/logs/ ; macOS ~/Library/Application Support/SoundCraft/logs/ ; Windows %APPDATA%\SoundCraft\logs\ ). A start from a desktop menu or the Dock has no terminal, so this file is what to attach to a bug report: the engine's panic report, audio devices that failed to open or broke, plugins that refused their stored state, CLAP plugins' own messages and files that failed to open land there. Each launch moves the previous log to soundcraft.1.log (and that one to soundcraft.2.log), so the log of a run that crashed survives the next start. The file stops growing at 16 MiB. --version writes no file, and runs with SOUNDCRAFT_NO_PREFS log to standard error only.

Variable	Effect
RUST_LOG	Log levels for standard error and log file. Default: info for SoundCraft's own crates, warn fo everything else. env_logger-style directives replace that, e.g. RUST_LOG=debug , RUST_LOG=warn,soundcraft_mix=t or RUST_LOG=info,wgpu_core=war directive ending in * covers eve target starting with it (soundcraft*=debug).
SOUNDCRAFT_NO_PREFS	No preferences read or written a no log file (agents' test runs).

The realtime audio thread never writes a record itself: the logger keeps the first one it logs (without waiting or allocating), counts the rest, and the UI thread writes them. The logger is apps/soundcraft/src/logging.rs . The web app logs to the browser console instead.

For agents and scripts

Everything you can click is also a command with an id and JSON parameters, and the same commands are reachable from the menus, the keyboard, the command line, a JSON control channel and an MCP server. Programmatic calls never open dialogs, and session.inspect reports exactly what changed, so an agent can check its own work.

```
# Headless: load the demo, turn the kick down, a
soundcraft-cli run --demo \
  --cmd 'mix.volume={"track":"Kick","db":-6}' \
  --cmd 'mix.insert={"track":"Bass","plugin":"eq
  --bounce mix.wav

# Drive the running app.
soundcraft --demo --control 7801 &
soundcraft-cli app --port 7801 transport.play
soundcraft-cli app --port 7801 ui.screenshot '{"

# Give Claude a DAW.
claude mcp add soundcraft -- soundcraft-cli mcp

# Or keep the agent to some folders (reads and w
soundcraft-cli mcp --automation-read-root DIR --
```

See docs/control-protocol.md and docs/mcp.md . soundcraft-cli commands lists every command.

Open **SoundCraft › Session Audio Health** to check loaded audio availability, sample-rate mismatches and clip source bounds across all playlists, including alternate takes. The report offers a refresh button and suggests how to resolve each issue. Scripts can obtain the same JSON report with soundcraft-cli run --demo --cmd 'session.audio_health={}' . This checks loaded media, not files on disk, signal levels, plugins or routing.

How it's built

SoundCraft is a Cargo workspace of small crates with strict layering: nothing below the UI knows about egui, so the interface could be swapped for another one.

Crate	What it does
soundcraft-time	Timebases, tempo and meter maps, timecode, grid
soundcraft-audio-io	Audio file formats and waveform overviews
soundcraft-midi	Standard MIDI Files and MIDI operations
soundcraft-dsp	Plugins, offline processing, meters, loudness, FFT
soundcraft-model	The session document
soundcraft-mix	The mix engine (realtime and offline share it)
soundcraft-engine	Commands, undo, editing, import/export, bounce
soundcraft-playback	Audio devices (CoreAudio, WASAPI, ALSA, WebAudio) and recording
soundcraft-automation	MCP server and control-channel client
soundcraft-ui-egui	The user interface

The code never panics on bad input: errors are values, and a malformed file or a bad agent call produces a message, never a crash. cargo xtask ci runs formatting, clippy, the tests, the asset and layering checks and the WebAssembly build.

Status and roadmap

SoundCraft is **pre-alpha**: it covers 94 % of the incumbent's menu items (measured), about 70 % of its features by presence, and is roughly 40 % of the way to replacing it for real work: record, edit, mix and bounce work end to end, but choosing an audio interface and buffer size, recording MIDI from a controller and session interchange (AAF/OMF) are still missing, and those are the alpha checklist. Translations are missing too. The status, the stage and our effort estimates are in ROADMAP.md ; the full assessment is in docs/target-app-parity.md , the ranked list of what's missing in docs/gaps.md , and the menu-by-menu comparison in docs/parity-checklist.md . Bug reports and wish lists are very welcome, in the issues or on Discord.

Downloads

Download SoundCraft from GitHub: the latest release has every build listed below, and all releases have earlier versions and their notes. <ver> in the file names is the version number, and SHA256SUMS.txt lists a checksum for every file.

Windows

Build	Installer	Portable
x64 (64-bit Intel/AMD)	soundcraft-<ver>-windows-x64.msi	soundcraft-<ver>-windows-x64-portable.zip
arm64 (Snapdragon and other ARM PCs)	soundcraft-<ver>-windows-arm64.msi	soundcraft-<ver>-windows-arm64-portable.zip
x86 (32-bit)	soundcraft-<ver>-windows-x86.msi	soundcraft-<ver>-windows-x86-portable.zip

Installers and executables are code-signed.

If the app doesn't open on Windows: the desktop app initializes only DirectX 12 by default. Letting wgpu also create an OpenGL instance can crash some graphics drivers (AMD's atio6axx.dll) before the window appears, so the app would flash in Task Manager and quit. WGPU_BACKEND overrides the default for troubleshooting (for example dx12 or vulkan). In PowerShell, from the folder containing the executable:

```
$env:WGPU_BACKEND = "vulkan"
& .\soundcraft.exe
Remove-Item Env:WGPU_BACKEND
```

An explicit gl override can bring the driver crash back on affected systems. The macOS, Linux and web backend defaults are unchanged.

DirectX 12 shaders are compiled with FXC, which is part of Windows, so no dxcompiler.dll or dxil.dll is needed ("Parent device is lost" at start, #185). $env:WGPU_DX12_COMPILER = "dxc" (or "auto") selects wgpu's DXC handling again for troubleshooting.

macOS

Build	File	Notes
App, universal (Apple silicon + Intel)	soundcraft-<ver>-macos-universal.dmg	Signed and notarized
Command-line tool, universal	soundcraft-cli-<ver>-macos-universal.zip	Signed and notarized

Linux

Format	x86_64	aarch64 (A
AppImage	soundcraft-<ver>-linux-x86_64.AppImage	soundcraf <ver>-linu aarch64.Ap
Flatpak	soundcraft-<ver>-linux-x86_64.flatpak	soundcraf <ver>-linu aarch64.fl
Debian/Ubuntu	soundcraft-<ver>-linux-	soundcraf <ver>-linu

	x86_64 .deb	aarch64.deb
RedhatRHEL/openSUSE	sounds-arf-noarch-linux-x86_64.rpm	sounds-arf-noarch-linux-aarch64.rpm
Tarball	sounds-arf-noarch-linux-x86_64.tar.gz	sounds-arf-noarch-linux-aarch64.tar.gz

RDC 0 (rsccx64): `sounds-arf-noarch-linux-rsccx64.tar.gz` A tarball only (glibc 2.39+, e.g. Ubuntu 24.04).

FreeBSD

Build	File
x86_64	`sounds-arf-noarch-freebsd-x86_64.tar.gz`

Web (WebAssembly)

Build	File	Notes
Static site	`sounds-arf-web-noarch.zip`	Runs in a modern browser; host it on any static server

The Crafting Apps

SoundCraft is one of the **Crafting Apps**: free, open-source creative tools for Linux, macOS, and Windows, written from scratch in Rust and each aiming to stand on its own.

App	What it's for	Code	Learn more
PhotoCraft	Image editing: layers, masks, filters, and PSD files	GitHub	Website
VectorCraft	Vector illustration	GitHub	Website
FilmCraft	Video editing: color and sound	GitHub	Website
LightCraft	Photo library and raw development	GitHub	Website
PdfCraft	Reading, organizing, and protecting PDFs	GitHub	Website
EffectCraft	Motion graphics and visual effects	GitHub	Website
DesignCraft	Page layout and publishing	GitHub	Website
SoundCraft	Recording, editing, and mixing audio - you are here	GitHub	Website

And ArtCraft itself, our AI image and video studio for artists who want real control.

JOIN US ON DISCORD

Come make things with us

Our Discord is where artists of every kind hang out: people who make music, film, art, and who care about tools that help them figuring out what they like to make. Share what you're working on, ask for help, tell us what's broken, or tell us what you wish these tools could do. Whatever your medium and however long you've been at it, you're welcome here.

discord.gg/yqcfqmzk9f · artcraft.info · The Crafting Apps · SoundCraft

License and credits

SoundCraft is dual-licensed under MIT or Apache-2.0, at your option. Copyright (c) 2026 ArtCraft Team and the SoundCraft contributors. Required notices are in NOTICE.

Bundled fonts, icons, images and other assets keep their own open licenses; each one is listed with its author, source and license in ATTRIBUTIONS.md.

Every icon in the interface is drawn in code, the app icon is original, and the demo song and the screenshots' audio are generated by the app itself. We don't ship third-party samples or artwork from anyone else in this repository.

The ArtCraft name, wordmark and logos in docs/brand/ are trademarks of the ArtCraft Team and are not covered by this license. They may be used only unmodified, and only as part of this repository and SoundCraft, under docs/brand/LICENSE_brand.txt. Forks and modified versions must remove them.

Made by the ArtCraft team and community.

Tracks and Groups, Clips, MIDI Editor, Memory Locations, and Undo history use movable panels. Edit and Mix retain separate arrangements; Clips and MIDI Editor belong to Edit, Memory Locations belongs to Mix. Drag a panel header to dock it into existing tools. Drag a panel header to a target edge or another header. Its context menu offers Float panel, Move to group, and Close panel. Existing panel toggles reopen panels at their saved location. Reset Panel Layout restores the app layout. Floating panels remain inside the current app window.

© 2026 GitHub, Inc.