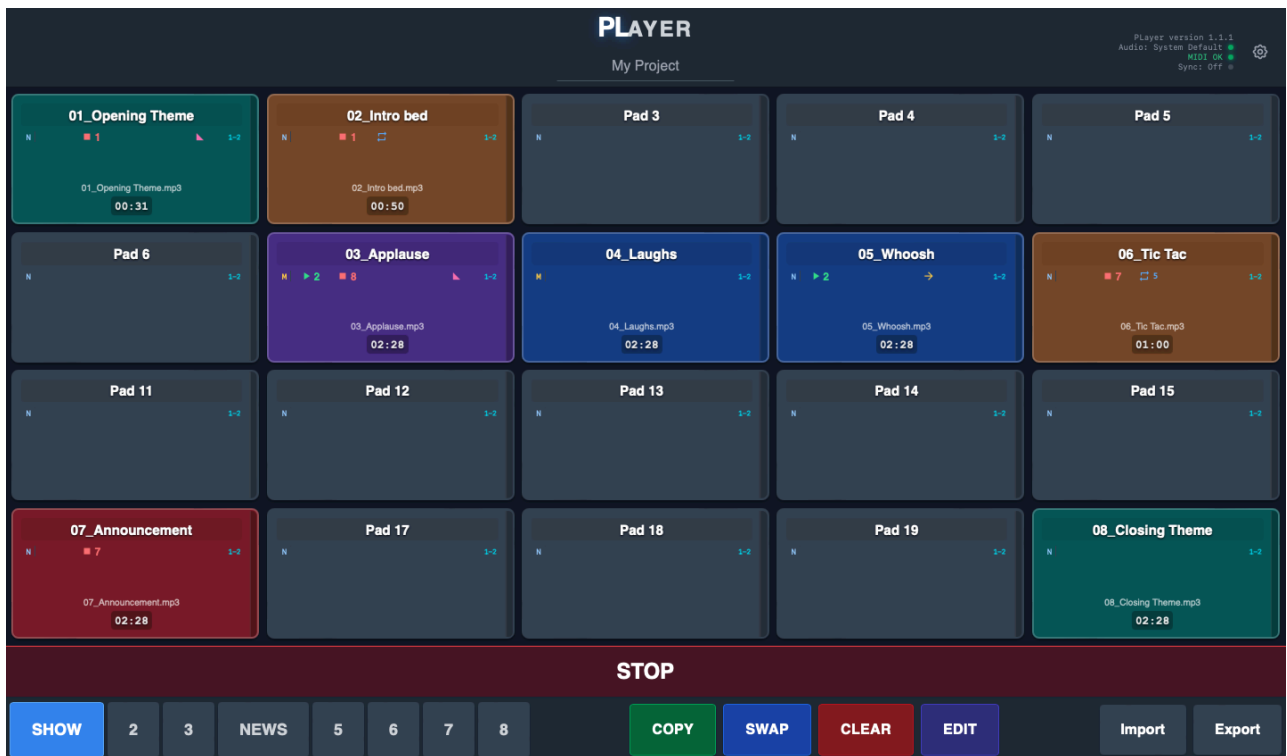


PLAYER

The Next-Level Audio Sampler

Version 2.0.0



User Manual

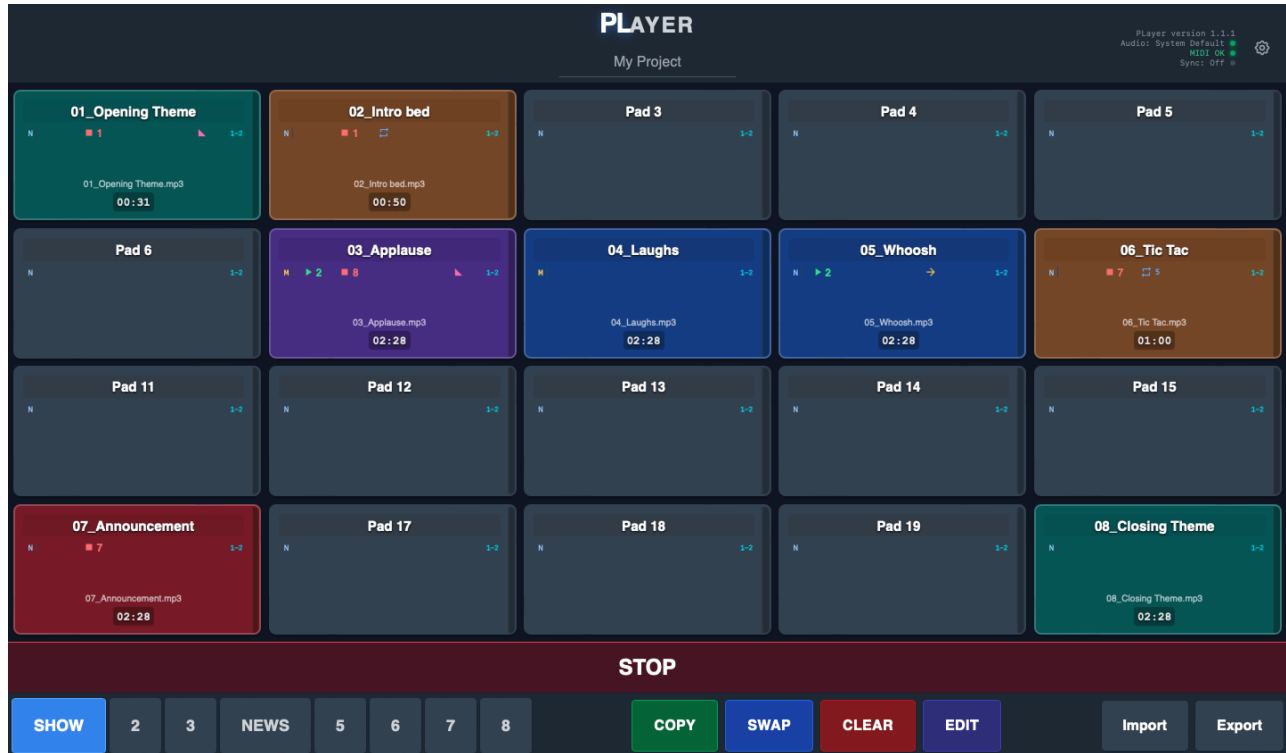
© 2026 Pierre Laurens Audio — All Rights Reserved
contact: pierre.laurens.audio@gmail.com

Table of Contents

1. Overview: Sound Pads — Main Window
2. System Configuration
3. Edit-Pad Window
4. Importing Audio Files
5. Waveform View: Cues, Fades & Transient Detection
6. Parametric Equalizer (EQ)
7. 3D Spatial Panner
8. Compressor
9. Reverb
10. MIDI Out
11. Export & Import XML Project Files
12. Recovery System
- A. Keyboard Shortcuts Reference
- B. Companion / Stream Deck Integration
- C. Redundancy Sync System

1. Overview: Sound Pads — Main Window

PPlayer is a next-generation audio sampler designed for professional live sound, theater, broadcast, and immersive audio workflows. Unlike legacy samplers limited to stereo playback and basic triggering, PPlayer provides a complete production toolkit: immersive 3D spatial panning (up to 7.1.4 Immersive format), per-pad parametric EQ, dynamics processing, reverb, multichannel smart import, hardware redundancy sync, and native MIDI/GPIO control — all in a single, latency-optimized application.



1.1 Pad Grid Layout

The main workspace is a **5-column × 4-row grid** of sound pads, giving you **20 pads per page**. With **8 pages** available, PPlayer offers a total of **160 independent sound pads**, each with its own audio file, volume, pitch, effects chain, spatial position, routing, and trigger configuration.

The grid dynamically resizes to fit your window. Each pad is a self-contained unit displaying real-time status at a glance.

1.2 Anatomy of a Sound Pad

Every pad on the grid shows the following information:

Element	Description
Pad Name	Custom or default name (e.g., "SFX Thunder"), centered, truncated with ellipsis if too long.
Duration	Total playback duration in MM:SS format.

Element	Description
Progress Bar	Semi-transparent white bar scaling horizontally to show real-time playback position.
VU Meter	Vertical green bar on the right edge. Grows from bottom upward, turns orange > 65% and red > 90%.
MIDI Badge	Yellow badge showing "MIDI: 60" (assigned note) or "MIDI: --" if unassigned.
GPIO Badge	Orange badge showing GPIO assignment status.
Status Icons	Small indicators for launch mode, play/stop group, loop count, and output routing.
Background Color	User-selected from a palette of 8 colors for visual organization.
Playing State	Golden border with glow effect when actively playing.

[Screenshot: Close-up of a single pad showing all visual elements]

1.3 Triggering Sound Pads

Pads can be triggered in multiple ways, depending on the configured **Launch Mode** (see Chapter 3):

Launch Mode	Click Behavior	Typical Use Case
Normal	First click plays from Cue In; second click stops.	Standard one-shot playback.
Restart (Stutter)	Every click stops and immediately restarts from Cue In.	Sound effects, Drums pad, stingers, stutter DJ effects.
Momentary (Gate)	Plays while mouse/key is held; stops on release.	Applause, crowd noise, sustained risers.
Pause	First click plays; second click pauses. Next click resumes from paused position.	Long narrations, blind-test, music tracks with pause/resume.

PRO TIP: In Momentary mode, PPlayer detects both MIDI Note On and Note Off events, making it perfect for keyboard-controlled gate effects.

1.4 Page Navigation

Eight page buttons are displayed at the bottom of the main window. Click any page button to switch instantly — all pads on the new page are ready to trigger with zero latency because audio data is kept loaded in the RAM.

You can **right-click** any page button to rename it (e.g., "ACT 1", "SFX", "MUSIC"). Page names are saved with your project XML.

PRO TIP: Use page names to organize your show: one page per act, one for music, one for SFX, one for ambiances. With 160 pads, you have room for the most complex productions.

1.6 Footer Bar



The footer toolbar provides special interaction modes:


Mode	Activation	Behavior
Edit Mode	Click the Edit button (indigo)	Click any pad to open its Edit-Pad window. Useful on touch devices.
Clear Mode	Click the Clear button (red)	Click any pad to remove its audio content.
Copy Mode	Click the Copy button (green)	Click a source pad, then click a destination pad to duplicate all settings.
Swap Mode	Click the Swap button (blue)	Click two pads to exchange their contents.
Export	Click the Export button	Open a system window to save the XML project file (name of the file is pre-filled with the project name in the header of the PLayer Main-window)
Import	Click the Import button	Open a system window letting you load an XML project file

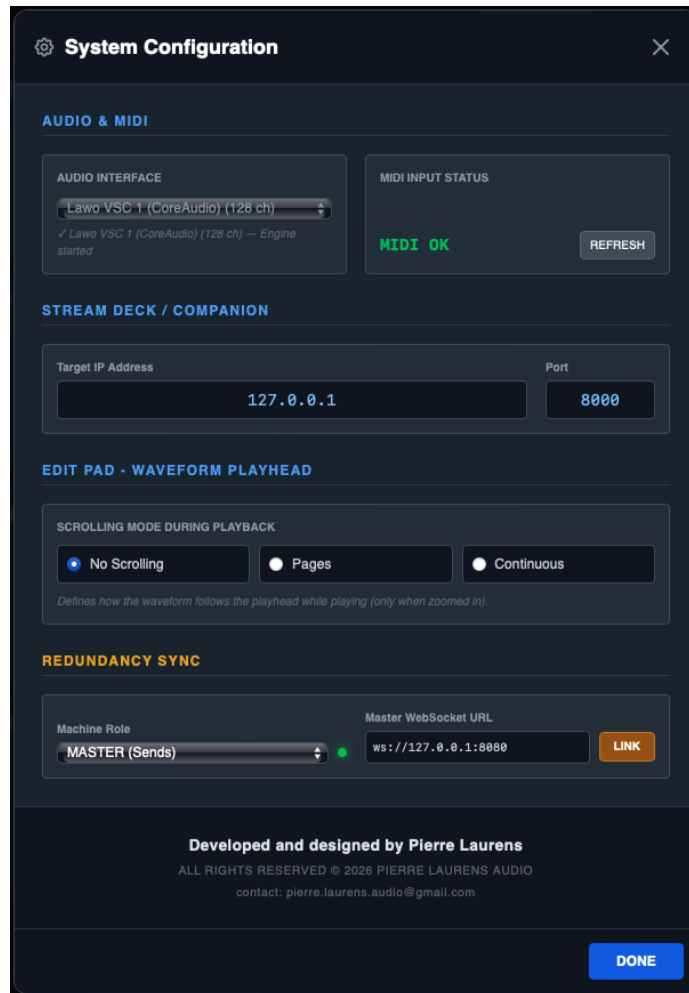
1.5 Header Bar

The header bar at the top of the window provides quick-access status indicators and controls:

Element	Description
Audio Status	Shows the active audio interface name and connection state.
MIDI Status	Green dot when MIDI is connected; gray when offline.
System Config Button	Opens the System Configuration modal (gear icon).

2. System Configuration

Access the System Configuration panel by clicking the **gear icon** in the header bar . This modal centralizes all hardware and network settings.



2.1 Audio & MIDI

Audio Interface

The **Audio Interface** dropdown lists all output devices detected by the system. Select your preferred interface — this determines which physical outputs PLayer uses for playback and routing. The change takes effect immediately; no restart required.

PRO TIP: For professional use on Windows, select your dedicated ASIO audio interface (e.g., Dante Virtual Soundcard, RME, MOTU) rather than the WDM/WASAPI system default. This ensures low-latency, multi-output routing.

MIDI Input Status

Displays the current MIDI input state: **MIDI OK** (green) when at least one MIDI input device is detected, or **MIDI Off** (gray) when no device is found. Use the **REFRESH** button to rescan after connecting a new MIDI controller.

PLayer connects to **all available MIDI input ports** simultaneously. Each port is identified by index and name, allowing per-pad device filtering in the Edit-Pad MIDI section (see Chapter 3).

2.2 Stream Deck / Companion

PLayer integrates with **Bitfocous Companion** offering an easy to control with a third-party equipment like a StreamDeck. Enter the **Target IP Address** and **Port** of your Companion instance.

PLayer sends real-time variables to Companion via HTTP POST:

Variable	Format	Description
pad_[1-20]_name	String	Pad display name on the current page.
pad_[1-20]_color	Hex (#RRGGBB)	Pad background color.
pad_[1-20]_state	"True" / "False"	Whether the pad is currently playing.

PRO TIP: Map these variables to Stream Deck buttons to get real-time feedback: colors change during playback and labels show pad names. A pre-configured Companion project is available on the PLayer website for simple Stream Deck integration.

2.3 Waveform Playhead Scrolling

This setting controls how the waveform canvas in the Edit-Pad window follows the playhead during playback when **zoomed in** (zoom > 1x). Three modes are available, inspired by DAW conventions:

Mode	Behavior	Equivalent in Pro Tools
No Scrolling (default)	The waveform view stays fixed. The playhead moves through the visible window and may exit the screen.	Scroll After Playback.
Pages	When the playhead reaches the right edge of the visible window, the view jumps forward by one screen-width.	Page Scroll mode.
Continuous	The waveform continuously scrolls to keep the playhead centered in the view.	Continuous Scroll mode.

This setting is saved in your app local storage and persists across sessions.

2.4 Redundancy Sync

For mission-critical live shows, PLayer supports a **Master/Slave redundancy** architecture over WebSocket. Two machines run PLayer simultaneously: one as Master, one as Slave. If the Master fails, the operator can instantly switch to the Slave.

Role	Behavior
OFF (Standalone)	Default. No synchronization. The machine operates independently.

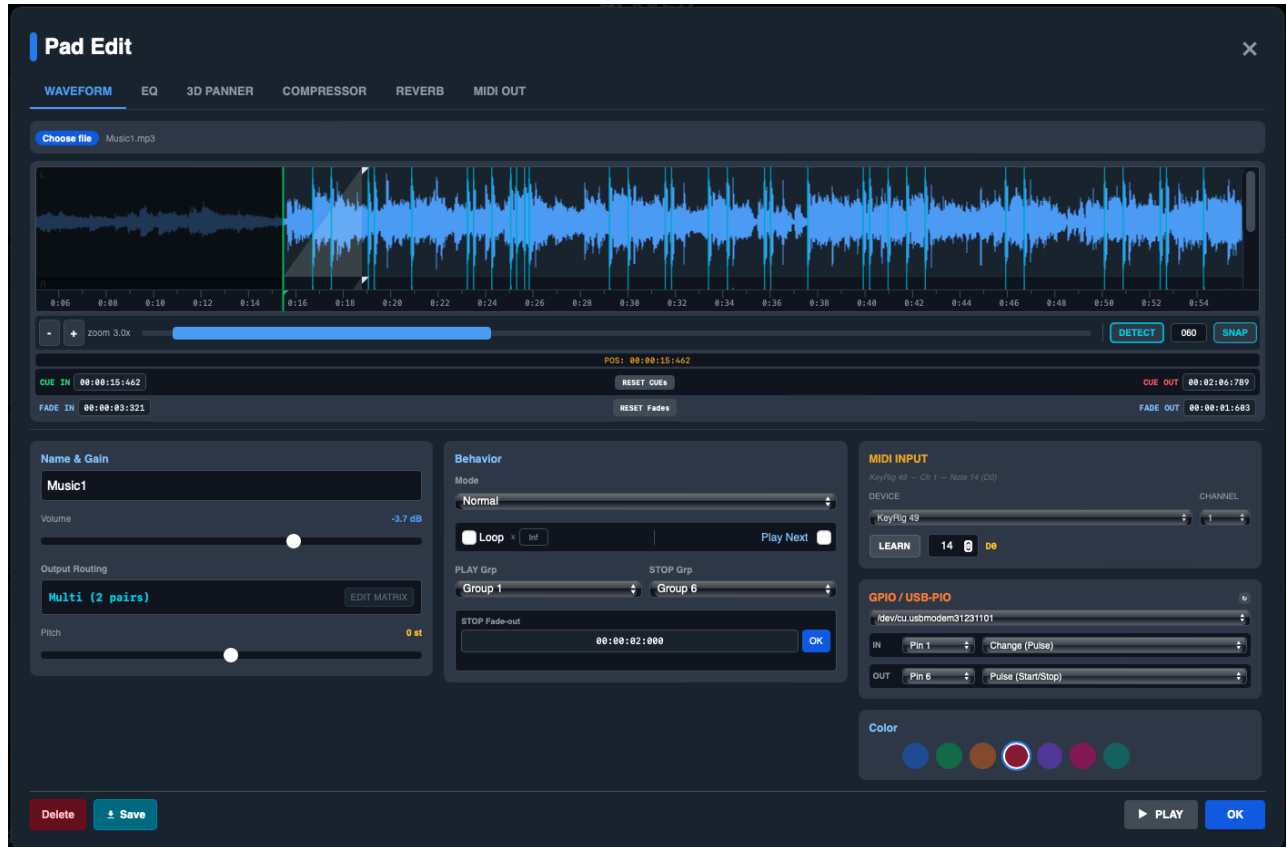
Role	Behavior
MASTER (Sends)	Broadcasts all pad triggers (play/stop), page changes, and state to connected Slaves.
SLAVE (Receives)	Receives and mirrors all actions from the Master in real-time via WebSocket.

Enter the **Master WebSocket URL** (e.g., ws://192.168.1.100:8080) and click **LINK** to establish the connection. The status indicator turns green when linked.

PRO TIP: For theater and broadcast, run two identical PPlayer instances with the same project XML loaded. Set one to MASTER and the other to SLAVE. If the Master machine crashes mid-show, the Slave has been tracking every action and is ready to take over instantly. For example, if you have a PAD playing or in Pause mode and your Master crashes, then you get your Slave with the exact same progression/state.

3. Edit-Pad Window

The Edit-Pad window is the command center for configuring every aspect of a sound pad. Access it by **right-clicking a pad** on the main grid or using **Edit Mode** on touch devices. The window is organized into tabs: **Sample**, **EQ**, **3D**, **Compressor**, **Reverb**, and **MIDI Out**.



3.1 File Import

Click the **Choose file** button to load an audio file. PPlayer’s engine supports:

Format	Extensions
Uncompressed	WAV, AIFF, PCM
Compressed (lossy)	MP3, OGG Vorbis, AAC
Compressed (lossless)	FLAC, ALAC

Upon loading, PPlayer decodes the file into raw PCM, generates multi-resolution waveform peaks and loads the audio into the RAM playback engine for zero-latency triggering.

PRO TIP: PPlayer keeps decoded PCM in the RAM. This means trigger latency is measured in microseconds, not milliseconds — critical for live theater cue-to-cue timing.

3.2 Custom Name

The **Name** text input at the top of the modal lets you assign a descriptive label (e.g., "Thunder Roll", "Applause Loop", "Act 2 Transition"). This name is displayed on the pad grid and sent to Companion/Stream Deck.

3.3 Volume

The volume slider ranges from **-60 dB** (effectively silent) to **+24 dB** (significant boost). The default is **0 dB** (unity gain). The value is applied as a linear gain multiplier in audio engine before any DSP processing.

PRO TIP: *Unlike consumer samplers that max out at 0 dB, PPlayer's +24 dB headroom lets you normalize quiet recordings without external processing. Combined with the compressor, you can achieve broadcast-level loudness per pad.*

3.4 Pitch

The pitch slider ranges from **-24 semitones** (two octaves down) to **+24 semitones** (two octaves up) in 1-semitone steps. Default is **0** (original pitch). Pitch shifting is achieved by adjusting the playback rate in the audio engine, which also affects duration.

3.5 Launch Mode

Selects the pad's triggering behavior. See section 1.3 for a full comparison table. The mode applies to all trigger sources: mouse click, MIDI, GPIO, F-key, and Companion.

3.6 Loop

Enable the **Loop** toggle to repeat playback between Cue In and Cue Out. The **Loop Count** field controls how many times the audio loops:

Loop Count	Behavior
0 or blank	Infinite loop — plays continuously until manually stopped.
1	Plays the Cue In→Cue Out region once (no repeat).
2 – N	Plays the region N times, then stops automatically.

PRO TIP: *For ambient loops in theater: set Loop Count to 0 (infinite), then use the Stop Delay (Fade Out) field to program a graceful fade when you stop the pad.*

3.7 Play Group

Assign pads to one of **8 play groups** (or None). When a pad in a play group is triggered from the **main grid**:

1. All other members of the same play group that are currently playing are stopped.
2. All members of the group are started simultaneously via a single atomic Rust IPC call (`play_group``), ensuring sample-accurate sync.

This is designed for synchronized multichannel playback:

1. **Stem Playback:** Launching backing tracks and click tracks in perfect sync.
2. **Immersive Sound Design:** Assigning the L, R, C, LFE, Ls, and Rs stems of a 5.1 mix to six pads within the same group. A single trigger fires all six components simultaneously.

PRO TIP: When previewing from the Edit-Pad window (spacebar or Play button), the play group is intentionally bypassed. Only the selected pad plays, so you can audition individual stems without triggering the entire group.

3.8 Stop Group

Assign pads to one of **8 stop groups** (or None). When any pad in a stop group starts playing, all other pads in the same stop group are stopped. This is useful for mutually exclusive sound sources (e.g., "Scene 1 Ambience" and "Scene 2 Ambience" should never overlap).

3.9 Play Next

When enabled, the **Play Next** toggle causes the next non-empty pad on the page to automatically start playing when the current pad finishes. This creates a sequential playlist within a page.

3.10 Stop Delay (Fade Out)

The **Stop Delay** field (in MM:SS.mmm format) defines a fade-out duration applied when the pad is stopped. Instead of an abrupt cut, the audio fades to silence over the specified time. Range: 0 (instant stop) to the pad's total duration.

3.11 MIDI Input

Each pad can be assigned to a specific MIDI note for hardware triggering. The MIDI Input section provides:

Control	Description
LEARN Button	Activates MIDI learn mode (pulses red). Press any key on your MIDI controller to capture the note, channel, and device automatically.
Device Selector	Dropdown of all detected MIDI input ports. Set to "Any" to respond to all devices, or select a specific controller.
Channel Selector	1–16 or "Any". Filters incoming MIDI messages by channel.
Note Input	0–127. Can be typed manually or captured via MIDI Learn.
Note Name Display	Shows the musical note name (e.g., C4, F#3) next to the note number.

The **Device Info Badge** at the top of the section always shows the currently learned configuration.

PRO TIP: Unlike most samplers that only support MIDI Learn, PPlayer lets you type note numbers manually. This is invaluable when programming a show without the physical MIDI controller present — you can set up all mappings from the studio.

PRO TIP: PPlayer identifies MIDI messages by device, channel, AND note. This means you can assign the same note number on two different controllers to two different pads — something most samplers cannot do.

3.12 GPIO / USB-PIO

For hardware button panels and custom trigger boxes, PPlayer supports **GPIO triggering via USB serial ports** (e.g., Arduino, Raspberry Pi Pico, custom PIO boards). Select the serial port,

configure the pin and trigger behavior (rise/fall edge). You can buy the official PLayer GPIObox on the PLayer website.

3.13 Color

A palette of **8 color swatches** lets you visually organize pads. Click a swatch to apply it. Colors are displayed as the pad's background on the main grid and sent to Companion for Stream Deck button coloring.

3.14 Output Routing Matrix

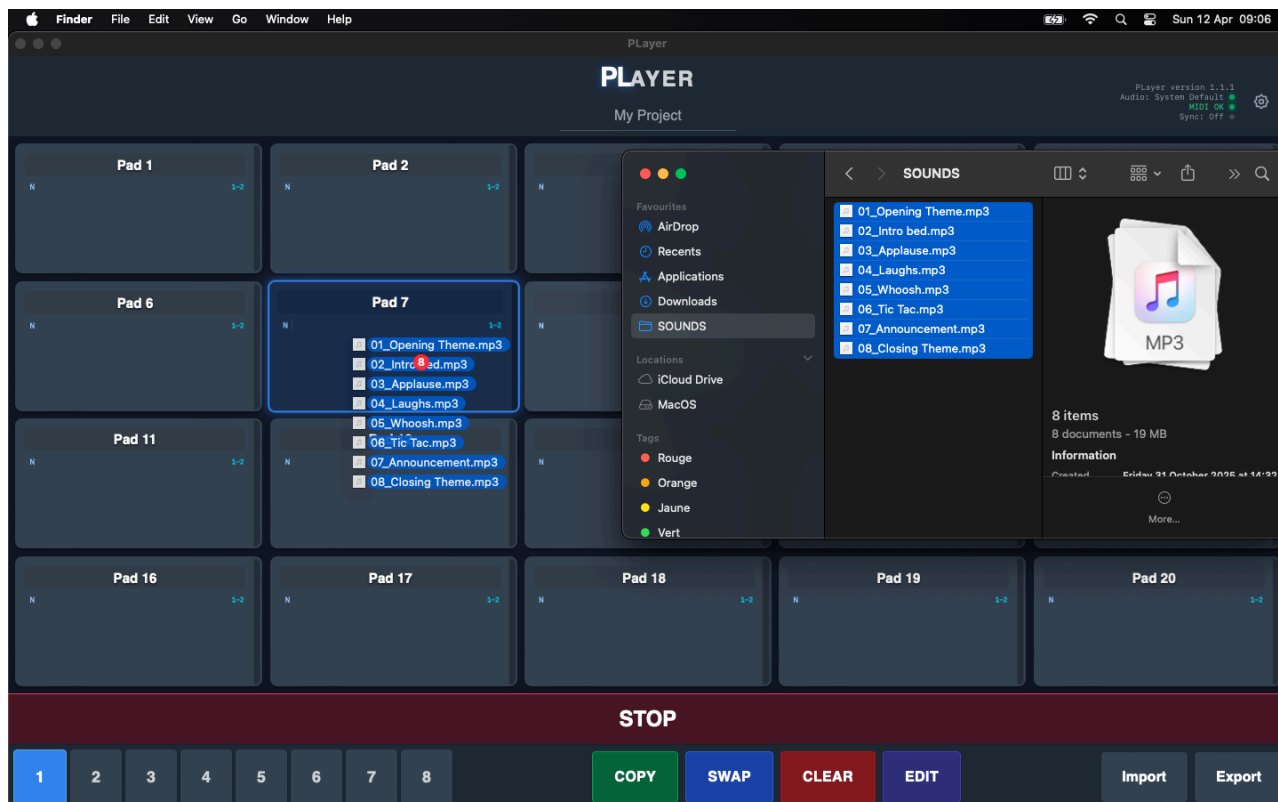
The routing matrix lets you assign the pad's audio to specific output pairs on your audio interface. This is configured via the output routing "EDIT MATRIX" button and is especially powerful when combined with the 3D Spatial Panner (see Chapter 7).

4. Importing Audio Files (Drag & Drop)

PPlayer supports intuitive drag-and-drop import directly from your operating system's file manager (Finder on macOS, Explorer on Windows).

4.2 Files Import

Select one or multiple files in your file manager and drag them onto the grid.



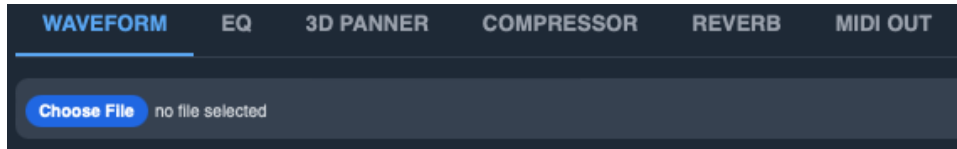
PPlayer uses a **smart fill algorithm**:

1. Starting from the pad you dropped onto, files are assigned to **empty pads only**, moving left-to-right, top-to-bottom.
2. Occupied pads are moved from left-to-right and top-to-bottom. Existing pad content and configurations are **never overwritten and are fully preserved**.
3. If there are more files than empty pads, a warning is displayed with the number of files that could not be imported.
4. If the import exceeds available space, existing pads at the end of the grid will be **shifted out and discarded**. A warning will indicate how many items are affected before you confirm.

PRO TIP: Prepare your audio files in numbered order (01_Intro.wav, 02_Scene1.wav...) and drag the entire folder selection onto pad 1. PPlayer fills the page sequentially, ready for a linear show.

Note: Please keep in mind that importing uncompressed immersive audio files may take longer. You might see an "Audio preparation..." notification while the system finishes loading the audio into the RAM and generating the waveform.

You can also load an audio file directly from the **Edit Pad** window by clicking the **'Choose File'** button located in the **Waveform** tab, just above the waveform view.

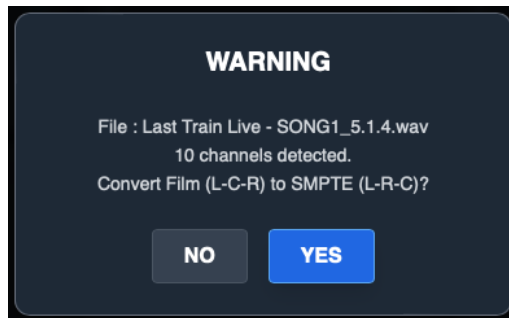


4.3 Smart Multichannel Import

This is one of PLayer’s most powerful features for immersive audio workflows. When you drop a **multichannel audio file** (e.g., an 8-channel WAV), PLayer detects it and presents a smart import dialog:

Channel Order Detection

Multichannel files from different sources use different channel orders. PLayer asks you to specify:

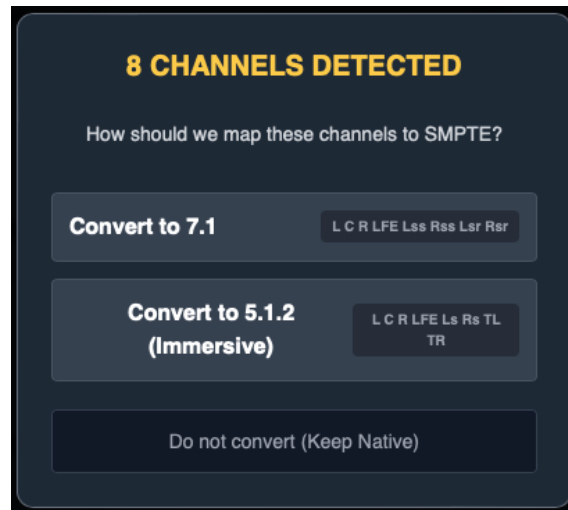


Standard	Channel Order	Common Source
FILM	L, C, R, Ls, Rs, LFE, ...	Film dubbing stages, cinema mixes.
SMPTE / ITU	L, R, C, LFE, Ls, Rs, ...	Broadcast, music, most DAW exports.

Format Mapping

Based on the number of channels, PLayer offers format mapping options:

Channels	Available Formats
6 channels	5.1 SMPTE, 5.1 FILM
8 channels	7.1 SMPTE, 5.1.2 (with height channels)
10 channels	7.1.2
12 channels	7.1.4 (Dolby Atmos bed)



Each channel is then shuffled to be automatically assigned to the correct output.

5. Waveform View: Cues, Fades & Transient Detection

The waveform canvas is the central visual element of the Edit-Pad window.



5.1 Waveform Display

Multichannel files are displayed with **one waveform lane per channel**, stacked vertically. For stereo files, you see two lanes (L/R); for 5.1, six lanes; etc. Background colors alternate between dark shades for visual separation.

The **ruler** at the top shows a time scale that adapts to the current zoom level: minutes and seconds at low zoom, down to individual milliseconds at high zoom.

5.2 Zoom & Scroll

Action	Result
Ctrl + Mouse Wheel	Zoom in/out centered on the mouse cursor position.
+ / – Buttons	Zoom in/out from the center of the current view.
Horizontal Scroll Bar	Scroll through the waveform when zoomed in.
Middle-click + Drag	Pan the waveform horizontally (hand tool).
Pinch Gesture (touch)	Zoom in/out on touch-enabled devices.

The **Zoom Level Display** shows the current magnification (e.g., "1x", "8x", "128x"). PPlayer supports zoom up to **500x**, where individual audio samples become visible.

5.3 Cue In / Cue Out Markers

Two draggable markers define the playback region:

Cue In (green line + triangle): The start point for playback. Default: 0 (beginning of file).

Cue Out (red line + triangle): The end point for playback. Default: end of file.

Drag the markers directly on the waveform or ruler, or type precise values in the **Cue In** and **Cue Out** time inputs (MM:SS.mmm format). The markers cannot cross each other. A **Reset** button restores them to the full file length.

When **Magnet Mode** is active, dragging cue markers snaps them to the nearest detected transient (see section 5.6).

5.4 Fade In / Fade Out

Fade handles appear as small **white triangles** at the top-left (fade-in) and top-right (fade-out) of the waveform, styled after Pro Tools. Drag them horizontally to adjust the fade duration. A shaded overlay visualizes the fade curve.

You can also type exact fade durations in the **Fade In** and **Fade Out** time inputs. Fades are applied as gain envelopes in the Rust engine at playback time.

5.5 Preview Cursor (Orange Dashed Line)

Click anywhere on the waveform to place the **preview cursor** (orange dashed line). When you press the **Play** button or **Spacebar** in the Edit-Pad window, playback starts from this position rather than the Cue In point.

The preview cursor works **anywhere on the timeline**, including before Cue In and after Cue Out. This lets you audition the full audio file regardless of the cue region.

5.6 Transient Detection & Magnet Mode

The **Detect** button runs a transient detection algorithm on the audio file. Detected transients appear as **cyan vertical lines** on the waveform. A **threshold slider** (0–100) controls detection sensitivity.

The **Magnet** button toggles snap-to-transient mode. When active, dragging cue markers or fade handles snaps them to the nearest transient — ensuring your edit points land precisely on beat boundaries or attack transients.

PRO TIP: For music cue editing, enable *Detect + Magnet* to snap your Cue In to the downbeat and Cue Out to the last note's release. This is dramatically faster than manual scrubbing.

5.7 Playhead

During playback, a **white vertical line** (with a small triangle indicator on the ruler) shows the real-time playback position. The **Cursor Info** display (top of the waveform) shows the precise time in MM:SS.mmm format, with a **red background** during playback and a **yellow background** when showing the preview cursor position.

6. Parametric Equalizer (EQ)

Each pad has a dedicated **7-band parametric EQ** with a fully interactive graphical interface. Access it via the **EQ** tab in the Edit-Pad window.



6.1 EQ Band Configuration

Band	Default Freq	Type	Default State
1	30 Hz	High-Pass Filter	Inactive
2	100 Hz	Low Shelf	Active
3	300 Hz	Peaking (Bell)	Active
4	1 kHz	Peaking (Bell)	Active
5	3 kHz	Peaking (Bell)	Active
6	8 kHz	High Shelf	Active
7	18 kHz	Low-Pass Filter	Inactive

Each band provides adjustable **Frequency** (20 Hz – 20 kHz), **Gain** (± 12 dB), and **Q** (bandwidth) for peaking bands. The high-pass and low-pass filters at bands 1 and 7 are inactive by default but can be enabled when needed.

6.2 EQ Canvas Interaction

The EQ canvas shows a logarithmic frequency scale (horizontal) and a dB gain scale (vertical). The resulting EQ curve is drawn in **yellow**. Each band is represented by a control point.

Gesture	Effect
Click + Drag (vertical)	Adjust gain of nearest band.
Click + Drag (horizontal)	Adjust frequency of nearest band.
Scroll Wheel over band	Fine-tune frequency.
Pinch (touch)	Adjust Q (bandwidth) of peaking bands.
Double-Click band	Reset band to default values.

6.3 EQ Bypass

The **Bypass** button disables all EQ processing without losing your settings. The visual indicator changes to show inactive state. This is useful for A/B comparison.

PRO TIP: *PPlayer applies EQ per-pad, not per-output bus. This means you can have 160 completely independent EQ curves running simultaneously — each pad has its own sound signature.*

7. 3D Spatial Panner

PPlayer's 3D Spatial Panner is one of its most advanced features, setting it apart from every other sampler on the market. It allows you to position audio sources in a full 3D space and render to immersive speaker configurations up to **7.1.4 Dolby Atmos**.



7.1 Output Format

Select the target speaker layout from the format dropdown:

Format	Channels	Use Case
Stereo	2 (L, R)	Standard playback, headphones.
5.1 SMPTE	6 (L, R, C, LFE, Ls, Rs)	Cinema, broadcast surround.
7.1 SMPTE	8 (L, R, C, LFE, Lss, Rss, Lsr, Rsr)	Extended surround.
5.1.2	8 (5.1 + Ltm, Rtm)	Immersive with top middle height.
5.1.4	10 (5.1 + Ltf, Rtf, Ltr, Rtr)	Immersive with 4 height channels.
7.1.4	12 (7.1 + Ltf, Rtf, Ltr, Rtr)	Full Dolby Atmos bed layout.

7.2 3D Visualization

The interactive 3D scene (powered by Three.js) shows speaker positions as **colored spheres** in a virtual room. The audio source is represented as a **draggable point** that you move through space.

PPlayer calculates distance-based gain attenuation for each speaker in real-time and sends the resulting gain matrix to the Rust engine for sample-accurate spatial rendering.

7.3 Controls & Shortcuts

Action	Effect
Mouse Drag a sphere	Move the audio source in the horizontal plane.
Click + CTRL + Mouse Drag a sphere	Adjust the audio source elevation (height).
Hold Shift + Click multiple sphere	Create a multi-selection to move in group.
3D ON/OFF Button	Enable/disable 3D processing. When OFF, audio uses standard routing.

Action	Effect
Alt + Click a sphere	Return the channel to its default position.
CTRL + Click a sphere	Mute the channel.

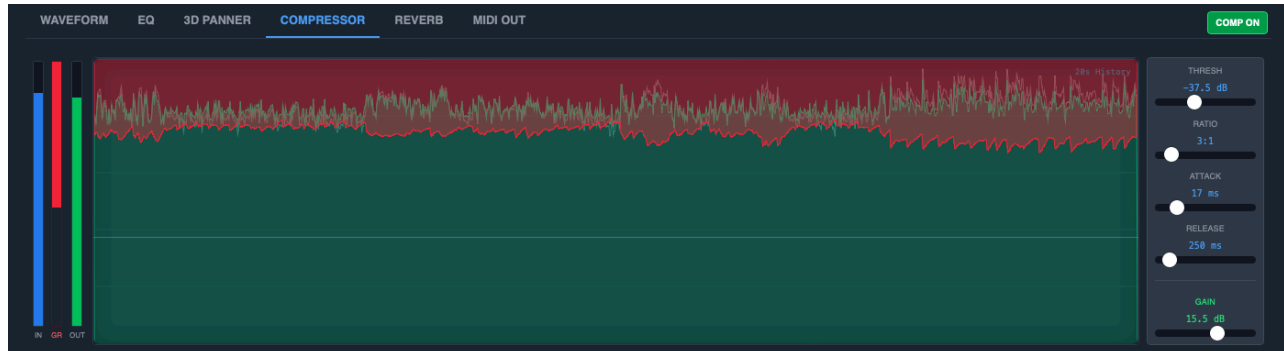
7.4 Use Case: Immersive Theater

Imagine a thunder sound effect for a theater production. With PPlayer's 3D Panner, you can position the thunder source above and behind the audience (elevation +45°, azimuth 180°). During playback, the Rust engine distributes the signal across your height speakers (Ltf, Rtf, Ltr, Rtr) and surround speakers (Lss, Rss) with physically modeled distance attenuation.

No other sampler offers this level of spatial control at the individual pad level. Traditional solutions require routing through an external Atmos renderer (Dolby Atmos Production Suite, L-ISA), adding complexity and latency. PPlayer integrates the spatial engine directly.

8. Compressor

Each pad has a dedicated **dynamics compressor** accessible via the **Compressor** tab in the Edit-Pad window. The compressor operates in the DSP chain after the EQ and before the reverb.



8.1 Parameters

Parameter	Default	Range	Description
Threshold	-24 dB	$-\infty$ to 0 dB	Level above which compression begins.
Ratio	4:1	1:1 to 20:1	Amount of gain reduction. Higher = more compression.
Attack	10 ms	0–100 ms	How quickly the compressor reacts to transients.
Release	250 ms	30–2500 ms	How quickly compression releases after the signal drops below threshold.
Knee	30 dB	0–60 dB	Smoothness of the transition into compression. Soft knee = gradual onset.
Makeup Gain	0 dB	0–24 dB	Post-compression gain boost to restore perceived loudness.

8.2 Visual Feedback

The compressor canvas shows a **transfer curve** (input level on X axis, output level on Y axis) with the threshold indicated as a horizontal red line. Three meters provide real-time feedback:

Input Meter (green): Incoming signal level before compression.

Output Meter (green): Signal level after compression and makeup gain.

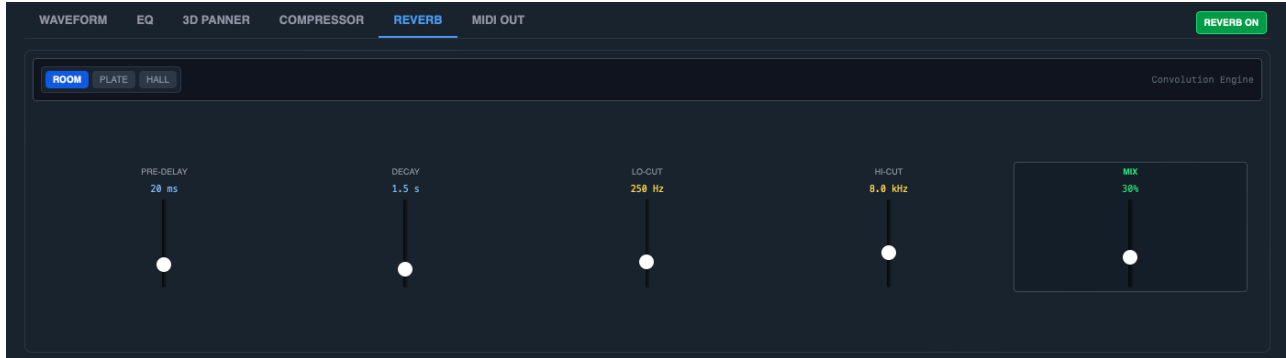
GR Meter (red): Gain Reduction amount — shows how many dB the compressor is reducing the signal.

8.3 Use Case: Broadcast Voiceover

For a voiceover pad that needs consistent level: set Threshold to -18 dB, Ratio to 3:1, Attack to 5 ms, Release to 200 ms, and Makeup Gain to +6 dB. The compressor tames dynamic peaks while the makeup gain brings the overall level up to broadcast standards. The visual GR meter confirms compression is active during loud passages.

9. Reverb

Each pad has an independent reverb processor accessible via the **Reverb** tab. The reverb operates at the end of the DSP chain (after EQ and Compressor).



9.1 Reverb Types

Type	Base Decay	Character	Use Case
ROOM	1.0 s	Tight, natural, short reflections.	Dialog, sound effects, intimate scenes.
PLATE	2.5 s	Bright, smooth, classic studio sound.	Vocals, stingers, announcements.
HALL	4.0 s	Spacious, deep, long tail.	Orchestral, ambient, dramatic moments.

9.2 Parameters

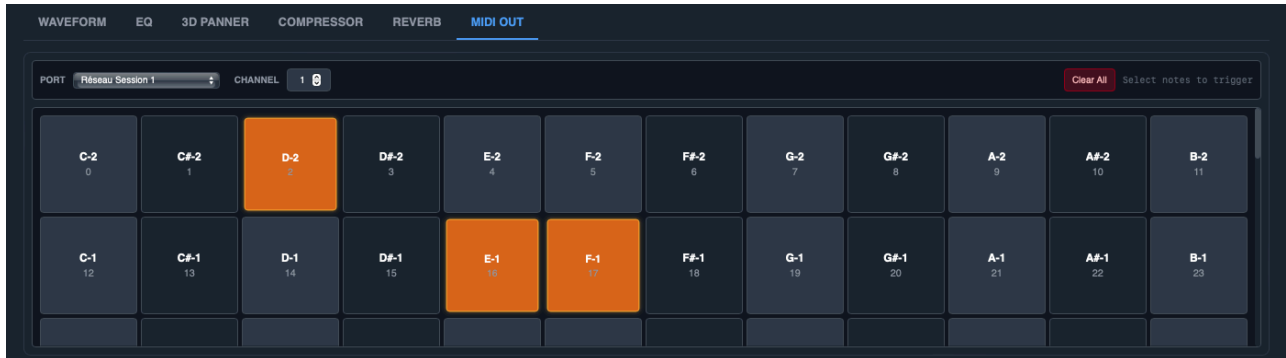
Parameter	Default	Range	Description
Pre-Delay	20 ms	0–100 ms	Time before reverb onset. Separates dry signal from reverb.
Decay	1.5 s	0.1–10.0 s	Length of the reverb tail.
Low Cut	50 Hz	20–500 Hz	High-pass filter on reverb input. Prevents muddy low end.
High Cut	8000 Hz	1–20 kHz	Low-pass filter on reverb input. Controls brightness.
Mix (Wet/Dry)	30%	0–100%	Balance between original sound and reverb.

The reverb operates with **tail mode** enabled: even after the pad stops, the reverb tail continues to decay naturally rather than being abruptly cut.

PRO TIP: In theater, use Room reverb with 15% Mix and 30 ms Pre-Delay on dialog pads to add a subtle sense of space without smearing intelligibility. For dramatic stingers, switch to Hall with 50% Mix for maximum impact.

10. MIDI Out

The **MIDI Out** tab lets each pad send MIDI Note On/Off messages when triggered, turning PPlayer into a MIDI controller as well as a sampler.



10.1 Configuration

Control	Description
Output Port	Select from detected MIDI output ports. Use "Default Output" for the system default.
Channel	MIDI channel 1–16 for outgoing messages.
Note Grid	128-cell grid (C-1 to G9). Click cells to toggle notes on/off. Active cells glow amber.
Clear Button	Deactivates all selected notes.

10.2 Behavior

When the pad starts playing, PPlayer sends a **Note On** message (velocity 100) for every active note in the grid. When the pad stops, PPlayer sends the corresponding **Note Off** messages.

Multiple notes can be active simultaneously, allowing you to trigger chords or multiple actions on external equipment.

PRO TIP: Use MIDI Out to trigger lighting cues on your lighting console, video playback on QLab, or effects on external hardware processors — all synchronized to your audio cues. This eliminates the need for a separate MIDI sequencer in your show control chain.

11. Export & Import XML Project Files

PPlayer saves and loads entire projects as **XML files**, providing a human-readable, portable project format.

11.1 Export

Click the **Export** button in the header bar (or use the keyboard shortcut). PPlayer generates an XML file containing:

1. All 160 pad configurations (names, volumes, pitches, cue points, fades, launch modes, groups, colors, MIDI assignments, GPIO assignments).
2. All DSP settings per pad (EQ bands, compressor parameters, reverb settings, 3D panner positions).
3. Output routing matrices.
4. Page names.
5. **Embedded audio data** in Base64 encoding (for pads loaded from raw data), or **file path references** (for pads loaded from disk in the Tauri desktop app).

A **progress bar** is displayed during export for large projects with many pads.

PRO TIP: *For backup and version control, export your project before each rehearsal. The XML format is text-based and works well with Git for tracking changes between show versions.*

11.2 Import

Click the **Import** button and select a previously exported XML file. PPlayer parses the file and reconstructs all pad states, settings, and audio data. A progress overlay shows the loading status.

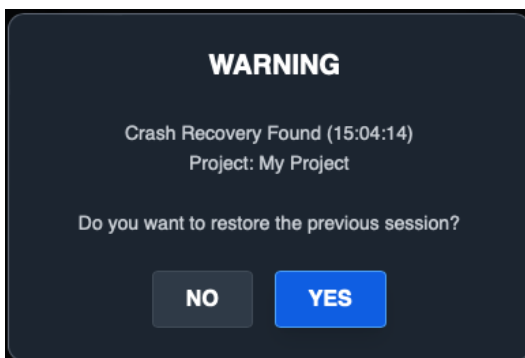
Import is additive: it replaces the current project entirely. Make sure to export your current project first if you want to preserve it.

11.3 Compatibility Notes

The XML format is forward-compatible: PPlayer gracefully ignores unknown tags from newer versions. This means you can load a project from an older version without losing data.

12. Recovery System

PPlayer includes an **automatic Crash Recovery System** designed for the reality of live production: unexpected power outages, system crashes, and accidental shutdowns.



12.1 How It Works

PPlayer continuously saves your project state in the background:

1. Every pad modification triggers an auto-save with a **30-second debounce** (to avoid excessive writes during rapid editing).
2. The complete project state is stored, including pad configurations, DSP settings, waveform peak data, and audio content references.
3. An undo/redo history stack (up to **30 steps**) is maintained.

12.2 Recovery on Startup

If PPlayer detects that the previous session did not close cleanly (e.g., the app was killed, the computer lost power), it presents a recovery prompt on the next launch:

"Do you want to restore previous session?" — Click **Yes** to restore all pads, settings, and audio references from the last auto-save point. Click **No** to start fresh.

After recovery, audio buffers are reloaded into the Rust engine from disk (if file paths are available) or from the stored Base64 data. This process is automatic and transparent.

PRO TIP: *For the highest reliability in mission-critical shows, combine the Recovery System with the Redundancy Sync feature (Chapter 2.4). Even if the Master machine loses power mid-show, the Slave machine has been tracking every action and is ready to take over. When the Master reboots, Recovery restores its state automatically.*

Appendix A: Keyboard Shortcuts

Shortcut	Context	Action
F1 – F12	Main Grid	Trigger pads 1–12 on the current page (launch mode dependent).
F1 – F12 (release)	Main Grid	Stop pad if in Momentary mode.
Spacebar	Edit-Pad Window	Play / Stop the active pad preview.
Ctrl+Z / Cmd+Z	Global	Undo last action.
Ctrl+Shift+Z / Cmd+Shift+Z	Global	Redo last undone action.
Ctrl+Y / Cmd+Y	Global	Redo (alternative shortcut).
Ctrl + Mouse Wheel	Waveform Canvas	Zoom in/out centered on cursor.
Enter	Text Inputs	Validate and apply the current input value.
Escape	Rename Dialog	Cancel page rename.
Right-Click	Pad	Open Edit-Pad window.
Right-Click	Page Button	Rename page.

Appendix B: Companion / Stream Deck Integration

PPlayer integrates with **Bitfocus Companion**, the open-source control software that powers Elgato Stream Deck, X-Keys, and dozens of other hardware control surfaces.

B.1 Setup

1. Install and launch **Bitfocus Companion** on the same machine (or a networked machine).
2. In PPlayer's System Configuration, enter the **IP Address** and **Port** of your Companion instance (default: 127.0.0.1:8000).
3. In Companion, create a **Generic HTTP** connection pointing to PPlayer's variables.
4. Map Companion buttons to PPlayer pad variables for visual feedback.

B.2 Variable Reference

PPlayer sends the following variables to Companion in real-time:

Variable Name	Type	Updates When
pad_[1-20]_name	String	Pad name changes (edit or project load).
pad_[1-20]_color	String (hex)	Pad color changes.
pad_[1-20]_state	"True" / "False"	Pad starts or stops playing.

Variables are updated on every pad edit, play/stop event, and project load. On startup and project import, all variables are reset to defaults.

Appendix C: Redundancy Sync System

The Redundancy Sync system provides **hardware-level failover** for mission-critical live productions. It synchronizes two PPlayer instances over a local network via WebSocket.

C.1 Architecture

The system uses a simple **Master/Slave** topology:

Component	Role
Master	Originates all actions (play, stop, page changes). Broadcasts state to Slaves.
Slave	Receives and mirrors all Master actions in real-time. Cannot initiate actions locally while linked.
WebSocket	Transport protocol. Low-latency, full-duplex communication.

C.2 Setup

1. Load the **same project XML** on both machines.
2. On Machine A (Master): System Config → Redundancy Sync → Set role to **MASTER**. Note the WebSocket URL shown.
3. On Machine B (Slave): System Config → Redundancy Sync → Set role to **SLAVE**. Enter Machine A's WebSocket URL. Click **LINK**.
4. The status indicator turns **green** when the connection is established.

C.3 Failover Procedure

If the Master machine fails during a show:

1. The Slave detects the WebSocket disconnection.
2. The operator switches the Slave's role from SLAVE to **OFF** (Standalone) in System Config.
3. The Slave now operates independently with the full, synchronized project state.
4. All pads, volumes, effects, and routing are identical to the Master's last known state.

C.4 Synchronized Data

The following events are synchronized from Master to Slave:

Event Type	Data Sent
Pad Play	Page, pad index, velocity, start position.
Pad Stop	Page, pad index.
Page Change	New active page index.
Parameter Changes	Volume, pitch, effects updates.

PRO TIP: For maximum reliability, connect both machines to the same network switch with a dedicated VLAN. WebSocket latency on a local gigabit network is typically < 1 ms, ensuring the Slave is always within one audio frame of the Master.