piPalette

Next generation workflow for the Polaroid ProPalette 8000 film recorder.

User Manual

Revision 0.1 • May 2026

Contents

Rolls

A **roll** is the basic unit of work in piPalette: an ordered queue of frames to expose against one film stock. Every roll is paired with one film table (`.FLM`) at the moment it is created, and a copy of that FLM is stored inside the roll. Subsequent edits or deletions in the **Film Tables** library never affect rolls that already exist — the film recipe a roll was created with is the recipe it will be exposed with.

> The "Rolls" page is what you will spend most of your time in. Creating a roll, queuing up images, adjusting per-frame settings, and starting the exposure run are all roll-level actions.

Creating a roll

Open the **Rolls** page from the sidebar and click **New roll** in the top-right of the panel. A small dialog asks for two pieces of information:

Name Anything that helps you find the roll later — a customer name, an order number, a date. It is not exposed onto the film and you can rename a roll at any time after creation.

Film table The FLM the roll will be exposed against. Only film tables already present in the library can be selected; if the list is empty, upload an FLM on the **Film Tables** page first. The dropdown shows each entry as `Name — Camera type · Color / B&W` so you can match the stock to the format you intend to shoot.

If the selected film table is black-and-white, a third option appears:

B&W filter The colour channel used for the single-pass exposure of a B&W stock — **Green**, **Red**, or **Blue**. Green is the usual default and is what most monochrome stocks were tested against; pick another channel only if the film stock or the look you are after asks for it.

Click **Create**. The new roll appears at the top of the list, immediately ready to receive frames.

> **The film table is locked the moment a roll is created.** piPalette snapshots the FLM bytes into the roll's folder. From this point on, the roll is self-contained: deleting the film table from the library, or editing its curves, has no effect on the roll. If you need different behaviour you must create a new roll.

The roll list

Each card on the **Rolls** page summarises one roll:

- The **name** you gave it, with a status badge to the right (*active*, *done*, *archived*).
- The film recipe — profile name, camera type, and colour mode. B&W rolls also show the chosen filter in parentheses.
- Pill counters showing the total frame count and, where non-zero, the number of frames that are `done`, `failed`, or currently `exposing`.

Click any card to open the roll's detail view.

Roll detail

The detail page is divided into two panels.

The **header panel** restates the roll's identity — profile, camera type, colour mode, aspect ratio, frame count, and the disk footprint of the roll (images, output renders, and thumbnails combined). The breadcrumb in the top-left links back to the roll list.

Below the header sits the **frames panel**, where you queue and arrange the images that make up the roll.

Renaming a roll

Click the roll title in the header. The text becomes editable in place. Type a new name and press **Enter**, or click away, to save. Press **Esc** to discard the change.

Deleting a roll

Use **Delete roll** in the top-right of the header. A confirmation dialog appears summarising what will be removed; deletion is permanent and also discards all uploaded source images, rendered output PNGs, and thumbnails that belong to the roll. The snapshot FLM is removed as well — your library copy is untouched.

Roll-wide options

A single roll-wide control lives under the header:

Skip recalibration between continuous frames When unchecked (the default), the recorder runs a calibration cycle before each frame. When checked, calibration runs only on the first frame of an uninterrupted sequence. Any pause, stop, or hardware error forces the next resumed frame back through a full calibration. Skipping recalibration is faster but, depending on the film stock and ambient conditions, can produce subtly inconsistent exposures across a long roll.

Changes apply instantly — there is no **Save** button.

Adding frames

The **Frames** panel has a dotted **dropzone** along the top. There are two equivalent ways to add images:

- Drag one or more files from your file manager and drop them onto the zone. The zone highlights amber while files are being dragged over it.
- Click the dropzone to open the system file picker.

Supported formats are JPEG, PNG, TIFF, and BMP. For multi-file uploads a progress modal appears that streams one file at a time, showing the current filename, the running count (e.g. *3 of 17*), and a progress bar. Each upload can be cancelled mid-batch; files already finished remain.

When an image lands in the roll, piPalette:

1. Saves the original under the roll's folder so the source is preserved.
2. Renders it through the FLM at the roll's aspect ratio to produce an exposure-ready PNG.
3. Generates a small JPEG thumbnail for the grid.

The new frame is appended to the end of the queue.

Frame settings

Each frame is a card in the grid. The card shows the thumbnail, the original filename, the source pixel dimensions, and the current per-frame settings. Below those are the controls you can adjust:

Resolution **4K** or **8K**. Defaults to 8K when the source image's long edge is 6000 pixels or more, otherwise 4K. Doubling the resolution roughly quadruples the exposure time and the output PNG size; on a 4K master use 4K to save time without losing fidelity.

Transform **Fit** preserves the entire source image and pads to the FLM's aspect ratio with the background colour. **Fill** crops the source so it covers the full frame without padding. **1:1** places the source at its native pixel size — no resampling. If the source is larger than the canvas in either dimension it is centre-cropped, and the card shows an amber warning explaining what was cropped. The aspect ratio of the frame itself is fixed by the FLM; you choose how the source is mapped into it.

Rotation Click the curved-arrow icon to rotate the source 90° clockwise. Each click advances through 0°, 90°, 180°, 270° and back to 0°. The thumbnail updates immediately.

Background Black or white — the colour piPalette pads with under **Fit**, and the colour the recorder uses around the image area. The recorder firmware only supports these two values.

Any change to a per-frame setting triggers a re-render of the exposure PNG on the server. The frame card dims and shows an amber spinner while this is happening — typically a fraction of a second for 4K and a few seconds for 8K. The thumbnail refreshes when the re-render completes.

Reordering and removing frames

Drag a frame by its body to reorder. The drop target is highlighted in amber, and the ordinal in the top-left of each card (`01`, `02`, …) updates immediately on release.

To remove a single frame, click the trash icon on its card. A confirmation dialog appears with the filename and warns that the upload and its renders will be deleted.

Skipping a frame

Use the **X** icon (top of the icon row on a pending frame) to mark the frame as **skipped**. Skipped frames stay in the roll for reference but the runner walks past them during roll runs. They are visually muted (greyscaled thumbnail, dimmed card, italic *Skipped — won't be exposed in roll runs* label) so you can tell at a glance that they aren't part of the queue.

To bring a skipped frame back into the queue, click the **circled +** icon on the card. It returns to `pending`.

Skipping is for frames you want kept in the list but excluded for now — say, an alternate crop you might come back to. If you simply don't want the frame at all, use **Delete** instead.

Frame status

Each card carries a small coloured dot in the top-right corner and a matching vertical stripe along its left edge:

Indicator	Meaning
muted	`pending` — queued, not yet exposed
amber	`exposing` — currently on the recorder
green	`done` — successfully exposed at least once
red	`failed` — the last exposure attempt did not finish
dim grey	`skipped` — in the roll but excluded from roll runs

These statuses are reflected in the count pills on the roll list and update live during an exposure run.

Exposing

There are two ways to expose: one frame at a time from the per-frame button, or the whole queue in a sequential roll run.

Single-frame Expose

On every pending frame card there is a primary **Expose** button. Done and failed frames show **Re-expose** or **Retry** in its place. Either way, clicking opens a confirmation dialog — the recorder cannot detect whether film is actually loaded, so the dialog asks you to confirm that it is, with bold text. Cancelling closes the dialog and nothing happens. Confirming kicks off the exposure.

While a single frame is exposing, piPalette puts up a **modal** that blocks the rest of the page. The modal shows the active phase (*Setup*, *Calibrating*, *Exposing R/G/B*, *Finishing*), a progress bar, elapsed time, and an ETA. There is no cancel button: an exposure cannot be aborted mid-burst without wasting the frame of film, so the modal stays in place until the recorder reports the frame complete. A typical single-frame exposure takes 60–100 seconds at 4K.

Roll runs

The roll header has a primary **Start exposing** button (disabled when no frames are pending) and a ghost **Stop after current** button (hidden until a run is active). Starting opens the same film-loaded confirmation; once confirmed, the runner walks every pending frame in order, exposing each one through the recorder.

Roll runs are intentionally **non-blocking**. While a run is in progress you can navigate to other pages — create new rolls, edit film tables, look at the device status — and the runner keeps going on the server. The currently-exposing frame card carries a phase + progress overlay so it is obvious which frame is active, and a status banner under the roll header tracks the overall *N / M done* count.

Stop after current is the only halt option. piPalette will let the active frame finish (so no film is wasted) and then halt. The banner turns yellow and reads *Stopping — finishing current frame* until the current burst lands. Resuming is just **Start exposing** again, which picks up at the next pending frame. piPalette forces a full recalibration on the next start after a stop or error, regardless of the *Skip recalibration* setting.

When a run completes, the banner turns green: *Run complete — all frames exposed*.

Re-exposing exposed frames

A frame that is `done` is, from the runner's point of view, finished — it is no longer pending and won't be picked up by **Start exposing**. There are three ways to re-expose a frame that has already been done:

Re-expose one frame, now Click the **Re-expose** button on a done frame. It triggers a single-frame exposure of that frame against the next available film — same modal, same flow as the initial Expose.

Reset one frame for the next roll run Click the small refresh icon in the frame's icon row. The frame flips back to `pending` — it stays where it is in the queue, history (exposure count, last-exposed timestamp) is preserved, and the next **Start exposing** will include it. No exposure happens yet.

Reset every done frame for the next roll run The roll header gains a **Reset done frames** button whenever the roll has any done frames. Clicking it (with confirmation) flips every done frame in the roll back to `pending` — useful when you want to re-shoot the entire roll on fresh film. Skipped and failed frames are left alone.

In all three cases, the frame's render and source image stay on disk — piPalette just clears the *done* flag.

Film Tables

A **film table** (`.FLM`) is the recipe the ProPalette 8000 follows when exposing a frame. It describes a single film stock to the recorder — its colour curves, aspect ratio, whether it is monochrome or colour, the camera format it was authored for, and the default filter for single-pass black-and-white exposure. Each frame you expose is the combination of one image and one film table.

piPalette stores the FLMs you upload in a small library so that you can pick one when creating a roll, without having to manage files manually on disk.

> Film tables are produced by the film manufacturer — historically Polaroid — and occasionally by labs or individuals for custom film stocks. piPalette only **reads** FLMs; it does not author or edit them.

The library

Open **Film Tables** from the sidebar. The page is a single panel with two areas: a dropzone along the top for adding FLMs, and a list below of everything currently in the library. If the library is empty the list area shows *No film tables yet — upload an FLM to get started*.

Uploading an FLM

There are two ways to add a film table, both equivalent:

- Drag one or more `.FLM` files from your file manager onto the dropzone. The zone highlights amber while files are being dragged over it.
- Click the dropzone to open the system file picker. Hold **Cmd** or **Ctrl** to select multiple files at once.

On upload, piPalette validates each file by parsing it through the recorder driver. A successful parse reads the following metadata straight out of the FLM header:

Name The film stock's identifier, as written into the FLM by its author — for example `EKTA100` or `PLUSXPAN`. This is the name you will see when selecting a film table for a roll.

Camera type The format the FLM was authored for: 35mm, 6×4.5, 6×7, 4×5, or similar. This determines the aspect ratio of every frame exposed against the recipe.

Color / B&W Whether the film is monochrome or colour. Black-and-white FLMs also expose a default filter colour (Green, Red, or Blue) used during the single-pass B&W exposure path.

If a file is malformed, encrypted with an unrecognised key, or otherwise fails to parse, the upload is rejected with an error toast and the file is **not** stored. Other files in the same batch continue to upload.

> **Duplicates are quietly ignored.** piPalette identifies each film table by a hash of its data, so re-uploading a file already in the library is a no-op — no error, no second entry. If you have two copies of the same FLM under different filenames, only the first one's

> filename will be remembered.

The library list

Each row in the library shows, from left to right:

- The film table's name (from the FLM header).
- Its camera type.
- Its colour mode — `Color`, `B&W`, or `B&W (Green)` / `B&W (Red)` / `B&W (Blue)` for monochrome stocks that specify a default filter.
- The original filename the file was uploaded with, in muted text. This is purely informational; piPalette stores the file under its own content hash internally.

Rows are sorted in upload order, oldest first.

Deleting a film table

Click **Delete** on the right of any row. A confirmation dialog appears with the film table's name and reminds you that deletion only removes the file from the library.

> **Deleting a film table is safe with respect to existing rolls.** Every roll snapshots the FLM at the moment it is created (see the *Rolls* chapter). Removing a film table from the library will not alter, break, or invalidate any roll that was already created against it — those rolls keep their own copy of the file. The only consequence of deletion is that the film table is no longer available as an option when creating *new* rolls.

The reverse is also true: if you delete a roll, the library copy of its film table is untouched.

Authoring your own FLMs

piPalette does not include an FLM editor. If you need a custom film table — to retune a stock for a different look, or to add a stock that was never shipped with the recorder — author the file with an external tool that follows the 2-master FLM convention and upload the result here.

Device

The **Device** page is where you tell piPalette how to reach the ProPalette 8000, and where you can confirm at a glance that the recorder is alive and responding. It is the only page concerned with the connection itself — exposure, slot installation, and roll playback all happen from the *Rolls* page once a device is connected.

The page is laid out as three panels: a status panel at the top, a connection panel below it, and a scan-results panel that appears on demand.

Hardware or Mock

piPalette can talk to either a real ProPalette 8000 attached to the host or a fully in-process **mock device** that fakes every command and response. The mock lets you evaluate piPalette — explore the interface, build rolls, dry-run the workflow — without a recorder connected. You cannot expose film with it. The choice is made in the **Connection** panel under **Mode**:

Hardware piPalette opens a real transport to the recorder. The status panel at the top fills in with live identification and mode data, and the topbar pill shows the product and firmware. Use this for everything but development.

Mock piPalette runs against an in-process simulator. The status panel collapses to a single muted card that says so explicitly — piPalette deliberately does **not** display the mock's invented numbers as if they were real device readings. The topbar pill simply reads *Mock*.

Switching modes applies immediately; there is no Save button. The recorder's actual connection state — and any error from a failed open — surfaces in the hint line directly under the Mode toggle.

Target

When the mode is **Hardware**, a second field labelled **Target** appears below it. This is where you tell piPalette which physical interface to talk to:

- A bare number — `4`, `5`, `6` — means "SCSI ID *N* reached through scsi2pi's `s2pexec` binary". This is the usual answer when the recorder is connected via the PiSCSI v2 HAT.
- A path like `/dev/sg2` means "the SCSI generic device at this path". piPalette will dispatch through the kernel's SG_IO ioctl. Use this when the host has a real SCSI host bus adapter or, historically, a PCMCIA SCSI card.

You do not need to pick a transport explicitly — piPalette infers it from the value's shape.

The field can also be left blank if you would rather have piPalette find the recorder for you. See **Scan** below.

> **Changing the target re-opens the connection on the next status poll.** You will typically see the topbar pill flicker through *No device* briefly while the new transport is being probed, then settle into *ProPalette 8000 · fw NNN* once the recorder answers.

Scan

Scan, in the top-right of the Device page or in the topbar, asks piPalette to enumerate every interface it knows how to talk to and report any ProPalette 8000 it finds:

- Every `/dev/sg*` device on the host, probed with SG_IO.
- Every SCSI ID from 0 to 7, probed via `s2pexec` (only if the `s2pexec` binary is on the system `PATH`).

Each candidate gets a short INQUIRY command; only those that respond with the PP8K identification string `DP2SCSI` are included in the results. A discovery sweep typically takes a couple of seconds.

When the scan finishes, the **Scan results** panel slides into view with one row per find, showing the recorder's identification line and the transport/target it was reached on. Clicking a row writes that target into the **Target** field and applies it as the new connection — it is the one-click equivalent of typing the value in by hand.

If the scan turns up nothing, double-check that the recorder is powered on, terminated correctly, and visible on the bus.

Identification

In **Hardware** mode with a recorder connected, the upper-left card of the status panel shows what piPalette read out of the device's INQUIRY and MODE SENSE responses. These values do not change at runtime; they are properties of the unit you are talking to.

Product The device family — always `DP2SCSI`/`ProPalette 8000` on a real recorder, never customer-rewriteable.
Firmware The firmware revision string the unit reports. ProPalette 8000 firmware versions in the field are typically in the high 500s.
Revision The hardware revision reported by the unit.
Buffer The size, in kilobytes, of the recorder's internal scanline buffer. Real hardware reports around 2456 KB; the mock device reports 4096. piPalette uses this to pace large exposures and does not normally expose it to the user beyond this read-out.
Max res The largest horizontal × vertical resolution the unit will accept, reported by the firmware. This is the upper bound for *Resolution* in per-frame settings; piPalette will not let you queue a frame that the recorder cannot expose.

Mode

The lower-right card shows the recorder's current operating mode — what it would do *right now* if you asked it to expose. These values do change as you (or piPalette) issue MODE SELECT commands.

Active slot The film-table slot currently loaded into the recorder's working memory. piPalette manages slot installation for you when a roll runs, so this value will normally read `19` (the scratch slot used by the driver) outside of an exposure run.
Resolution The horizontal and vertical resolution the recorder is configured to expose at, in scanlines.
Camera back The recorder's reading of which camera back is mounted (35 mm, 6×4.5, 6×7, 4×5, and so on). For correct exposure the mounted back must match the camera type of the film table being used.
Luminance, Color balance, Exposure Three triplets (R, G, B) describing the per-channel in-

tensity, trim, and shutter time the recorder will apply on the next exposure. They are derived from the active film table; you do not edit them directly.

The topbar pill

Every page in piPalette carries a small connection pill in the top-right of the topbar. Its dot is green when piPalette is connected to a recorder (real or mock) and dark grey when it is not. The label to the right of the dot summarises the current connection:

- `Mock` — running against the simulator.
- `ProPalette 8000 · fw NNN` — connected to real hardware, with the firmware revision shown in monospace.
- `No device` — no connection; in **Hardware** mode the word *offline* in red on hover reveals the underlying error.

The refresh icon next to the pill forces a fresh status read, bypassing piPalette's two-second cache. Use it after physically reconnecting cables or power-cycling the recorder.

System

A fourth panel at the bottom of the Device page surfaces the host's piPalette install and offers one-click updates. On a managed install (one that came from `deploy/install.sh`) the panel shows:

Version The git tag piPalette is currently running from — for example `v0.1.0`. If the host is on a commit past the latest tag the value reads `vX.Y.Z-N-gSHA` (the standard `git describe` shape).
Commit The full commit hash, for the times you need to point at exactly what is deployed.
Install *Managed (systemd)* on a regular install, *Development* on a checkout you are hacking on directly.
Last update The status message left behind by the last update worker — *idle* after a clean run, an error string if something went wrong.

Checking for updates

Click **Check for updates** in the panel header. piPalette runs `git fetch --tags` against `origin` and reports back:

- If the latest tag is the one already running, the panel reads *Up to date — running vX.Y.Z*.
- If the tag on the remote is newer, the panel shows the new tag, lists the commits between your current `HEAD` and the new tag (newest first, capped at fifty), and offers an **Update now** button.

Applying an update

Update now opens a confirmation that names the target tag and warns that the service will restart — make sure no exposure is running. Confirming kicks the update flow:

1. piPalette writes the target tag to `/var/lib/pipalette/pending-update`.
2. The systemd unit `pipalette-update.service` runs as root via the sudoers entry the installer dropped.
3. The worker fetches, checks out the new tag, reinstalls Python dependencies (including any new pinned pp8k version), and restarts `pipalette.service`.

The browser stays on the page, watching `/api/version`. As soon as the service comes back

up reporting the new tag, the page reloads automatically.

> **Development checkouts cannot update via this button.** If the host is not a managed install — for example a `pip install -e .` from a clone — the **Update now** button is disabled and the panel reads *Development* under *Install*. Update by `git pull` and reinstalling as usual.