
GENERATOR

There are two distinct FLUX generators. They are not interchangeable.

PERSONAL GENERATOR

The personal generator is the core production tool that converts a session of original photographs into a complete FLUX issue, assigned to the permanent personal chronological archive.

Canonical photo count: 36 photographs per issue.

What it does:

1. Reads EXIF data from every photograph (timestamp, camera settings, GPS)
2. Renames files to the canonical filename format using EXIF timestamp
3. Assigns the next sequential issue number from the personal archive
4. Assigns frame numbers in capture-timestamp order
5. Generates the manifest CSV
6. Generates web derivatives (1400 px long edge, JPEG 70)
7. Generates thumbnails for the contact sheet
8. Generates the contact sheet PNG (6 × 6 grid)
9. Generates the PDF (44 pages: cover, protocol, 36 images, contact sheet, manifest, blank back)
10. Updates the archive: rebuilds timeline.json, issue page HTML, explore.html
11. Deploys to S3

What it does not do:

- Edit photographs
- Select photographs
- Reorder photographs
- Add captions
- Alter metadata

The generator is a translation machine. Originals in. Publication out.

Invocation:

```
python3 generate_flux_issue.py 305          # generate issue 305
python3 generate_flux_site.py              # rebuild site
python3 deploy_s3.py --bucket flux-dantesisofo --issues 305 --cloudfront E1QJZ6W1R67CZD
```

PUBLIC GENERATOR

The public generator is a separate tool accessible at:

<https://flux.dantesisofo.com/generator/>

It allows any photographer to run the FLUX protocol on their own photographs without requiring access to Dante Sisofo's personal archive or file system.

Status: Live. Launched 2026-05-12. Browser-based, no installation required.

Canonical photo count: 36 photographs per issue.

The 36-frame count is not a default – it is the standard. It is the same for personal issues and public issues.

Implementation:

- No account required
- Browser-based (HTML + jsPDF – runs entirely client-side, no server upload)
- Requires exactly 36 JPEG photographs
- EXIF timestamp read for chronological ordering and caption data
- GPS coordinates read from EXIF and surfaced in caption and manifest if present
- Protocol page included in every generated PDF
- Output format structurally identical to personal archive volumes
- No GPS requirement (GPS used when present, never required)

Image system:

- EXIF-aware compression via `createImageBitmap({ imageOrientation: "from-image" })` – portrait images stored as landscape with EXIF rotation tag are correctly oriented before processing
- `effectiveGridH` computed per-batch from minimum width-limited landscape height – adapts to 3:2, 4:3, or mixed batches automatically
- `IMG_SCALE = 0.98` applied to all drawn image dimensions – 2% optical reduction, centering anchor unchanged, adds ~2.2 mm breathing room per horizontal edge
- All 36 frames centered at optical page centre (4.25 in from top on an 8.5 in page)

Output difference from personal generator:

The public generator does not assign issue numbers from Dante Sisofo's personal chronological sequence. Public generator output is a standalone artifact – a complete FLUX issue, but not an entry in the personal archive.

FLUX QUEUE

Before photographs are formally assigned to a numbered issue, they enter the FLUX Queue.

The queue is a staging area inside `FLUX_ISSUES/FLUX_QUEUE/`. Photographs in the queue have been processed (renamed, derivatives generated, manifests built) but have not yet received an issue number.

The queue is visible in the public archive as an unnumbered stream of recent work. When the queue contains 36 photographs for a session, a new numbered issue is published.

Workflow:

1. Drop JPEGs into FLUX_UPLOAD/
 2. Run ./flux_update.command
 3. System automatically processes new photographs into the queue
 4. When 36 queue photographs exist, the system publishes the next issue
-

QR CODES IN GENERATED ISSUES

The QR code structure differs between personal archive issues and public generator issues.

PERSONAL ARCHIVE ISSUES – TWO QR CODES

QR 1 – Protocol page (p.03), bottom-right:

Target: <https://flux.dantesisofo.com/generator/>

QR 2 – Manifest page (p.43), bottom-left:

Target: https://flux.dantesisofo.com/issues/FLUX_NNN/

Personal issues carry both QRs. The protocol QR is a standing invitation to any reader. The manifest QR connects the physical object to its permanent digital record.

PUBLIC GENERATOR ISSUES – ONE QR CODE ONLY

QR 1 – Protocol page (p.03), bottom-right:

Target: <https://flux.dantesisofo.com/generator/>

The public generator embeds the protocol page QR only.

The manifest page (p.43) is present and contains the full frame list – timestamps, GPS coordinates, frame numbers. It is documentation. It does not carry a QR code.

Reason: The public archive infrastructure does not yet exist. Placeholder or optional manifest QR behavior – user-provided URLs, generator redirects, blank spaces – creates inconsistent printed artifacts and premature infrastructure complexity.

Public generator output = downloadable issue only.

No automatic publishing. No manifest QR.

FUTURE PUBLIC PUBLISHING INFRASTRUCTURE

The public generator is live. The next layer – public archive infrastructure – is not yet built.

When built:

- User generates issue via public generator
 - System creates a permanent public archive page for that issue
-

– Manifest QR is automatically generated, pointing to that page

The public archive would be searchable by photographer, date, and issue number, and chronologically ordered across all public contributors.

This infrastructure does not yet exist. When it does, the manifest QR for public issues becomes as locked and permanent as the protocol QR.

HISTORICAL FRAME COUNTS

The FLUX system has operated under three different frame-count logics. These are distinct implementations, not variants of the same protocol.

15 PHOTOGRAPHS – early public generator prototype

Context: Experimental specification for an early public generator concept.

Status: Superseded. Never built or deployed.

Not part of the canonical FLUX protocol.

This count was chosen arbitrarily as a lower barrier for public participation. It produced a 22-page PDF. The logic was never implemented in a working generator. The specification is documented here only for historical completeness.

50 PHOTOGRAPHS – historical Dante Sisofo archive batching

Context: The original personal archive queue logic. Dante Sisofo's first 304 issues were generated using this batch size.

Status: Historical. No new issues will be generated at 50 frames.

Not the canonical protocol – a production artifact of the early system.

This count was not derived from any photographic tradition or formal constraint. It was a working batch size chosen for the personal archive pipeline. It produced a 58-page PDF with a 10×5 contact sheet grid (50 cells for 50 frames). The grid was portrait-oriented, meaning landscape and portrait thumbnails rendered at different heights – the grid lock did not hold.

36 PHOTOGRAPHS – canonical FLUX issue

Context: Current and permanent standard for all FLUX issues – personal and public.

Status: Locked. All future generators use this count.

This is the protocol.

36 = one roll of 35mm film. The constraint is not arbitrary – it comes from the physical medium FLUX references. It produces a 44-page PDF, a 6×6 contact sheet grid (36 cells for 36 frames), and a clean spread sequence with no orphaned pages.

The 6×6 grid is the correct geometry for 36 frames. Both landscape and portrait thumbnails are height-limited in the cell, so all 36 frames render at the same height – the same grid lock that governs full-page image placement extends to the contact sheet.

NAMING CONVENTION

Personal archive:

FLUX_001 FLUX_002 ... FLUX_NNN

Canonical filename format:

YYYY-MM-DD_HH-MM-SS_PhotographerName_OriginalCameraFilename.JPG

Example:

2026-05-07_20-40-55_DanteSisofo_R0022598.JPG

Collaborative project filenames:

{project-slug}_NNN_{photographer-slug}_{YYYY-MM-DD}_{HH-MM-SS}.jpg

FLUX_WIKI_v1.1 – flux.dantesisofo.com/wiki/generator/