Quick Start — Blinder (v4.0)

BlindKit v4.0 — Quick Start (Blinder, **keys & audit**)

You manage the BLINDER repo on your laptop; the experimenter does not have access to it.

Environment (choose one)

Conda (recommended)

```
conda env create -f environment.yml
conda activate blindkit
```

Or: venv + pip

```
python -m venv venv && source venv/bin/activate
pip install -r requirements.txt
```

Initialize your repo (on your laptop)

Create only the blinder tree:

```
python blindkit_v4_0.py init-dual \
  --study-id STUDY_X \
  --blinder-root ./study_X_blinder \
  --only blinder
```

Register animals

```
python blindkit_v4_0.py register-animal \
  --blinder-root ./study_X_blinder \
  --animal-id RAT001 --sex F --weight 230g
# repeat per animal
```

Plans (reproducible with YYYYMMDD seeds)

Behavior (2×A, 2×B per animal)

```
python blindkit_v4_0.py plan-behavior \
  --blinder-root ./study_X_blinder \
  --date-seed 20250821 \
  --agents A B
```

Physiology (50/50 cohort) — legacy■aware

Legacy CSV: `animal,agent` (header optional)
Legacy JSON: `{"RAT001":"A",...}` or `{"assignments":{"RAT001":"A",...}}`

```
# Strict: fail if legacy has unknown animals
python blindkit_v4_0.py plan-physiology \
  --blinder-root ./study_X_blinder \
  --date-seed 20250821 \
  --agents A B \
  --legacy-csv ./legacy_phys.csv
# Lenient: allow legacy-only animals
```

```
python blindkit_v4_0.py plan-physiology \
  --blinder-root ./study_X_blinder \
  --date-seed 20250821 \
  --agents A B \
  --legacy-json ./legacy_phys.json \
  --allow-unregistered
```

Overlays / Labels

```
# Behavior (prompts for animal, session 1-4, base syringe ID)
python blindkit_v4_0.py overlay-behavior --blinder-root ./study_X_blinder

# Physiology (one per animal; echoes planned agent to console for blinder only)
python blindkit_v4_0.py overlay-physiology --blinder-root ./study_X_blinder

# Viral aliquot micro-label (cap/side code input)
python blindkit_v4_0.py overlay-aliquot --blinder-root ./study_X_blinder
```

- Text label files saved under `BLINDER/labels/` (+ optional QR PNGs if `qrcode` is installed).
- Registry updates: `BLINDER/labels/registry.json` (append-only).

Handoff & reconciliation

Experimenter logs receipts; you reconcile to mark overlays USED.

```
python blindkit_v4_0.py reconcile-usage \
  --blinder-root ./study_X_blinder \
  --experimenter-root ./study_X_experimenter
```

Anatomy blinding (to experimenter)

```
python blindkit_v4_0.py blind-anatomy \
  --blinder-root ./study_X_blinder \
  --experimenter-root ./study_X_experimenter \
  --input-root  /data/histo_unblinded \
  --allow-missing-index \
  --seal
```

Creates:

- BLINDER `configs/anatomy_crossref.json` & `configs/anatomy_blind_map.json`
- EXPERIMENTER `anatomy_blinded/` copies + `configs/anatomy_blinded_manifest.json`
- If `--seal`: BLINDER `archives/anatomy_blinded_.zip` (+ `.sha256`)

Post■hoc bundle (for review)

```
python blindkit_v4_0.py package-unblinding \
  --blinder-root ./study_X_blinder \
  --experimenter-root ./study_X_experimenter \
  --out ./study_X_unblinding_bundle.zip

# Anyone can verify integrity; also logs to specified root(s)
python blindkit_v4_0.py verify-posthoc --bundle ./study_X_unblinding_bundle.zip --blinder-root ./study_X
```

Audit log queries

```
# Last 30 actions in the blinder repo
```

```
python blindkit_v4_0.py audit-show --root ./study_X_blinder --tail 30

# Only overlays
python blindkit_v4_0.py audit-show --root ./study_X_blinder --action overlay-physiology
```

Repo hygiene

- Keep BLINDER and EXPERIMENTER as **separate** versioned repos under your respective control.
- Never put BLINDER secrets into the experimenter repo (configs/labels/archives).
- Commit after each command to timestamp the trail; consider adding a pre-commit hook later.