

Report Studio Developer Onboarding

Scaffold, build, and audit your first branded PDF

Task-oriented steps with terminal screenshots.
Dogfood from ReportLab — same CLI consumers use.

AUDIENCE Report authors · agent operators

SECTION I

Overview

Who this guide is for and what you will produce.

This guide walks through the report-studio CLI from a clean consumer checkout. Each section pairs explanatory text with a terminal screenshot — text first, image second per GitHub Docs and ScreenSnap 2026 conventions. You need uv, the nlt-report-studio pin, and write access under reports/.

Methodology

This brief is generated from the live NLT document suite (nlt-report-studio v0.2.0, brand nlt-v3.2). Diagrams and tables reflect build-time workspace discovery where noted.

Audience

- Investors & board — Vol 1-2
- Architects — Vol 3-4
- Engineering DD — Vol 5-8

Related proof

portfolio.nltlabs.ai · briefs.nltlabs.ai

SECTION II

Scaffold a report

Create reports//story.py from a YAML template manifest.

Follow the steps below, then confirm the scaffold on disk before editing narrative.

- Choose a template: `report-studio init --report my_brief --template portfolio_story`.
- Confirm `reports/my_brief/story.py` exports `build_story()`.
- Edit narrative using premium helpers — `section_header`, `premium_table`, `screenshot_figure`.
- Keep `path::symbol` citations for `report-studio verify`.

Report Studio

```
$ uv run report-studio init \  
  --report my_brief \  
  --template portfolio_story  
  
Wrote reports/my_brief/story.py  
Wrote reports/my_brief/__init__.py  
Merged pyproject dependency nlt-report-studio @ v0.2.0  
  
reports/  
  my_brief/  
    story.py      ← build_story()  
    __init__.py
```

Figure 1 — `report-studio init` scaffolds `story.py` and merges `pyproject` deps.

SECTION III

Build a PDF

Render a PDF from `build_story()` with brand `nlt-v3.2`.

Run `build` from the repo root so `brands/` and `templates/` resolve correctly.

- `uv run report-studio build --module reports.my_brief.story:build_story --out out.pdf`.
- Pass `--template` when the story expects template part titles.
- Set `--title` and `--subject` for PDF metadata and running headers.
- Ship PDFs via `apps/docs build` — never hand-edit binaries.

Report Studio

```
$ uv run report-studio build \  
  --brand nlt-v3.2 \  
  --template portfolio_story \  
  --module reports.my_brief.story:build_story \  
  --out dist/my-brief.pdf  
  
Building ... portfolio_story / nlt-v3.2  
Wrote dist/my-brief.pdf (842 KB)
```

Figure 2 — build command writing a branded PDF to disk.

SECTION IV

Audit and verify

Catch thin pages and broken links before merge.

Use `--audit` on build for density checks; verify guards `path::symbol` citations in stories.

- Add `--audit` to build for post-render density and link checks.
- `report-studio verify --builders reports/my_brief/story.py` for citations.
- Engine PRs: `uv run pytest tests/test_user_guide.py` and `tapps_validate_changed`.
- Ops volumes use `audit_profile ops`; briefs use `stakeholder` or `reference`.

Report Studio

```
$ uv run report-studio build \  
  --module reports.my_brief.story:build_story \  
  --out dist/my-brief.pdf \  
  --audit --audit-profile reference  
  
audit: 12 pages, outline ratio 0.14 ✓  
audit: link check 0 broken ✓  
audit: min alnum density per page ✓  
Build passed.
```

Figure 3 — --audit profile flags thin pages before deploy.

SECTION V

Next steps

Where to go after your first PDF.

- Read docs/CONSUMER_GUIDE.md for git-tag pinning in sibling repos.
- Use screenshot_figure() and scripts/ingest-screenshots.py for new captures.
- Pick templates via plugin skill /report-studio:pick-template.
- See docs/DOCUMENT_PLATFORM_ROADMAP.md for release notes and marketing briefs next.

Suite index

Platform briefs Vol 1-10 map to templates in docs/NLT_DOCS_SUITE.md. Document platform expansion adds user guides, release notes, and marketing briefs on the same report-studio CLI.

Confidential — prepared for NLT Labs stakeholders. Outward names follow docs/NLT_BRAND.md. Internal checkout names appear only in builder ops volumes (9-10).