Document Synthesiser (DocSyn)

Contributor Guide — Adding and Curating New Documents

How to ingest, triage, promote, clean, synthesise, and validate content into the SSOT — including brand-new domains.

Date: 09 September 2025 | Project: DocSyn

Contents

Placeholder for table of contents 0

1. Purpose & Principles

This guide shows exactly how to add documents to DocSyn while preserving the Single Source of Truth guarantees. It covers the common path (10th doc into existing blueprints) and the edge path (completely new domain).

- Promote via staging → curation → build – never edit SSOT directly.
- Surgical de-duplication removes global leakage; provenance is preserved.
- Human-in-the-loop: contributors propose content; CI gates and reviewers approve.

2. Where to Drop Your File

2.1 If you know the blueprint (recommended)

```
merge_pr/updated/<blueprint>/
  └── your-file-name.md
```

Example:

```
merge_pr/updated/documenter/api-style-guide-v4.md
```

2.2 If you're unsure of the blueprint

Use a triage area or drop at the root — promotion still works; reports will flag it as uncategorised.

```
merge_pr/updated/_triage/candidate-risk-controls-checklist.md
# or
merge_pr/updated/candidate-risk-controls-checklist.md
```

3. Minimal Front-Matter (Provenance)

Add a small YAML header at the top of the file. This makes provenance auditable and helps the Curator Agent learn.

```
---
source: Company Policy v3.1 (PDF)
source_url: https://example.com/policy_v3_1.pdf
retrieved: 2025-09-09
owner: Risk/Legal
tags: [policy, risk, controls]
---
```

4. Commands You'll Run (Promotion → Build)

```
# 1) Promote staged docs into curated blueprints (with backups)
python scripts/promote_updated.py

# 2) Clean, synthesise, validate (surgical de-dup + invariants)
```

```
make ci

# 3) Review artifacts
less dist/PROMOTION_REPORT.json
less dist/LEAKAGE_REPORT.json
less dist/VALIDATION.json
open dist/SSOT.md
```

If you don’t use make, run your one-click entrypoint instead:

```
python one_click_repair_and_build.py
```

5. Decision Tree — Where Does My Doc Go?

- Clearly blueprint-specific? → merge_pr/updated/<blueprint>/…
- Unsure which blueprint? → merge_pr/updated/_triage/… (or root)
- Clearly global (Part A)? → add part: A in front-matter and drop into _triage for curator routing.

6. Completely Different Documents (Brand-New Domains)

Use this path when the material does not fit any existing blueprint (e.g., a new function or product area).

6.1 Create a new blueprint

```
# New blueprint skeleton
blueprints/<new_blueprint>.md   # will be constructed from promoted content
merge_pr/updated/<new_blueprint>/new-topic-overview.md
merge_pr/updated/<new_blueprint>/runbook-v1.md
```

Then run promotion and build. The consolidator synthesises a clean datasheet from the promoted content and removes global leakage automatically.

6.2 If the content is Global (Part A)

If your new material adjusts the Router, Unified Risk & Control Model, or Standard Dev Environment, mark it as Part A and route through triage:

```
---
part: A
section: Router
source: Team Design Notes (md)
owner: Platform
---

# Proposed Router addition
```

```
...
```

A reviewer merges the best version into the single canonical Part A sections; CI ensures they appear only once in the SSOT.

6.3 New repository vs. federation (bigger expansions)

- If the corpus is large and distinct (e.g., a new business line), consider a separate repo with the same pipeline.
- Use federation later to reuse Part-A building blocks across repos and publish a unified catalog.
- Keep namespacing clean to avoid cross-domain contamination in retrieval.

7. Quality Gates You Can Trust

- De-duplication: heading-scoped + content fingerprints (escaped headings and tables handled).
- Leakage reports: structured blocks (e.g., Agent Query Index tables) are removed from blueprints; details in LEAKAGE_REPORT.json.
- Invariant checks: single copy of Part-A globals; code-fence parity; file size sanity; promotion safety nets with timestamped backups.

8. Reset or Clean Rebuild (Optional)

Normally unnecessary — the pipeline is idempotent — but if you want a fresh slate:

```
rm -rf dist/
python scripts/promote_updated.py
make ci
```

If promotion went wrong, restore from dist/backups/blueprints/<timestamp>/.

9. Examples

9.1 Existing blueprint (Documenter)

```
merge_pr/updated/documenter/api-style-guide-v4.md
---
source: Internal API Guide (GDoc)
owner: DevEx
tags: [api, style]
---
# API Style Guide v4
...
```

9.2 Unsure → triage

```
merge_pr/updated/_triage/candidate-risk-controls-checklist.md
---
source: Audit Deck (PPTX)
owner: Risk
---
# Risk Controls Checklist (Draft)
...
```

9.3 Completely new domain (Analytics)

```
merge_pr/updated/analytics/analytics-oncall-playbook.md
merge_pr/updated/analytics/kpi-definitions-v1.md
```

10. Troubleshooting — Fast Signals

- Validation says OK but you see duplicates → ensure your front-matter and headings are not blockquoted; re-run make ci and open LEAKAGE_REPORT.json.
- Promotion did nothing → check paths under merge_pr/updated/; see PROMOTION_REPORT.json for moved/ignored files.
- Root SSOT overwritten → expected: it is AUTOGENERATED. Always edit staged sources instead.
- Non-markdown sources → convert to .md first, keep headings/tables, add provenance front-matter.

One-Page Cheat Sheet — Add the 10th Document

- Drop here: merge_pr/updated/<blueprint>/ or _triage/
- Front-matter: source, URL, retrieved, owner, tags
- Promote: python scripts/promote_updated.py
- Build: make ci
- Review: dist/SSOT.md, LEAKAGE_REPORT.json, VALIDATION.json
- Reset (optional): rm -rf dist/ then promote + build

For brand-new domains: create merge_pr/updated/<new_blueprint>/... (or mark part: A for global changes), then run the same flow.