DOCUMENTATION PAGE

# First governed documentation contribution

# First governed documentation contribution

## Your First Safe Contribution

Treat the first contribution as a small, inspectable product change. Start at the repository root, read the onboarding notes and the sealed authority, and choose one documentation improvement that can be explained in one sentence. The safe result is a candidate page plus the records that let another reader trace its identity, sources, journey goals, and validation. It is not a claim that the contributor has approval, platform access, readiness, or a customer outcome.

Keep the contribution narrow enough that a reviewer can compare every material sentence with the three authority-backed source records. Do not fill a missing fact with plausible prose. If the source boundary, page identity, or intended audience is unclear, preserve the uncertainty in the candidate record and ask for the separately governed decision. A useful first task demonstrates reproducible product evidence and a clean handoff, rather than maximizing the amount of text changed.

## Repository Roles

This `governed-execution-engine-docs` repository is the Documentation Steward product workspace. `content/` holds semantic Markdown, `registries/pages.json` holds page identity and provenance metadata, and the `journeys/` record holds natural-language reader goals and acceptance traces. The candidate record in `releases/` describes the parked product handoff. `build/index.json` is a deterministic projection of product sources, so it is an output to inspect and never a source to edit.

Select the documentation-steward author role for this task: orient, author one bounded page, synchronize its registry entry, record reader goals, generate the projection, and stop at independent review. The separately governed employee repository and its employee, routing, validator, activation, approval, and publication controls are authority paths, not product files to edit here. The names of external repositories do not grant access or permission, and no unsupported details about their layout or operators should be inferred.

## Authority Boundaries

Open `sources/authority.json` as a sealed, read-only input. It has revision `2026-08-08`, sets `employee_may_modify` to `false`, and lists exactly `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`. Match each logical ID in `registries/sources.json`, confirming its locator, revision, digest, authority class, and `public-safe` disclosure before using it in the page header or registry. The page uses no claim references because no registered product claim is needed for this orientation guide.

The editable product records for this task are the new Markdown page, its one `registries/pages.json` entry, the candidate record, and the journey manifest. The authority file and source registry are read-only evidence; the governing repository is outside this workspace; and `build/` is generated. A missing, stale, conflicting, or non-public-safe source is an unresolved boundary, not a reason to edit authority, copy private governance material, invent a claim, or declare the task complete.

## Candidate-Only Workflow

First choose the page path and stable ID, then write the metadata block with the exact title, public-safe visibility, audience, reading modes, empty claim references, and the three sealed source references. Add the same identity and reference fields to `registries/pages.json`. Next create the four reader journeys in `journeys/first-task-guide-user-journeys.json`: each journey should state a natural-language goal, name its author authority, remain non-exploratory, and trace one of AC-1 through AC-4. Keep the candidate release record synchronized with the page, workstream, evidence locations, and candidate-only boundary.

Run the declared deterministic product path from the repository root: `python3 tools/validate_content.py`, followed by `python3 tools/build_projection.py`. Inspect

`build/index.json` for the new page ID, path, title, visibility, claim references, source references, content revision, and authority identity. Fix semantic sources when a check exposes a gap; do not patch the generated projection. The candidate remains parked for review even when these product checks pass, and no release pointer, live directory, promotion, publication, schedule, rollback, or activation follows from this workflow.

## Review Handoff

Hand the candidate package to an independent reader with the Markdown page, the synchronized page registry entry, the journey manifest, the sealed source references, the candidate record, and the generated projection location. Ask the reader to follow each natural-language goal and preserve the exact command, exit status, observed output, and any unresolved blocker. A declared command in a manifest is an allowlist declaration, not evidence that a platform ran it; the author must not turn an expected result into an attestation.

After the reader record is available, route the package to an independent reviewer on a separate review route. The reviewer checks scope, provenance, heading and metadata requirements, page-registry synchronization, journey coverage, and the candidate-only boundary. Keep the reviewer’s verdict and findings separate from the author’s summary. If reader or review evidence is missing or fails, retain that fact and leave the candidate parked; do not narrow the journey, delete the failed evidence, self-certify a pass, or publish.

## Acceptance Checks

- **AC-1 — Repository orientation:** The contributor identifies this repository as the Documentation Steward product workspace, explains the roles of `content/`, `registries/`, `journeys/`, `releases/`, and generated `build/`, selects the documentation-steward author role, and leaves unsupported details about the separately governed repository unresolved.
- **AC-2 — Authority boundaries:** The contributor reads the sealed authority, verifies revision `2026-08-08`, `employee_may_modify: false`, and the exact three public-safe source IDs, then uses only those IDs for page provenance without changing authority or copying private governance material.

- **AC-3 — Compliant candidate creation:** The contributor creates one narrow candidate-only page with synchronized identity fields, the named audience, appropriate reading modes, only authority-backed references, four traced journey goals, and a deterministic projection generated from source files.

- **AC-4 — Review handoff:** The contributor packages the candidate for an independent reader and an independent reviewer, keeps product conclusions separate from platform attestations, preserves missing or failed evidence, and confirms that no approval, promotion, publication, scheduling, rollback, release-pointer, activation, or live-directory action occurred.

---

**Provenance:** registry-listed (no candidate packet)
**Registry Source:** `content/first-governed-documentation-contribution.md`