

DOCUMENTATION PAGE

Pre-authoring Reader-Journey Contract Preflight

Pre-authoring Reader-Journey Contract Preflight

Purpose

Run this preflight before an auto-orch workflow author drafts semantic content or an operator lets a governed execution consume budget. It turns the sealed authority record and immutable reader-journey manifest into a small, actionable contract decision: either every required fact, input, command, and reference is current and the workflow may proceed, or the result names the exact unresolved field or path that must be repaired. The result is a product-side conclusion; it does not change authority, certify a platform route or validator, or grant approval, activation, promotion, publication, scheduling, rollback, or live directory authority.

Inputs Checked

Start with `artifacts/preflight/source-authority-seal.json`. Confirm its sealed authority revision, authority-listed source IDs, and the permitted public-safe reference metadata for each source in `registries/sources.json`. The current source IDs are `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`; their locator, revision, digest, authority class, and disclosure are inputs to the comparison, not prose to be inferred. Then read the immutable `artifacts/preflight/pre-authoring-reader-journeys.json`: its contract path, four journeys, acceptance criteria, source references, traces, expected results, and `command_allowlist` define what this preflight must cover.

Resolve every repository-relative reference named by those records. That includes

`sources/authority.json` , `registries/sources.json` , this page at `content/public/pre-authoring-reader-journey-preflight.md` , and the declared product-contract command target `tests/test_product_contract.py` . Check that the page metadata and its single `registries/pages.json` record agree on page ID, title, visibility, audiences, reading modes, claim references, and source references. Treat the manifest as immutable and do not use `build/` , `releases/` , `publication/` , private governance material, or generated output as authority or as a substitute for a missing input.

Actionable Errors

Report an error with its category, repository-relative location, observed value, expected value, and bounded repair target. For authority errors, name the source ID and field when a locator, revision, digest, authority class, or `public-safe` disclosure is missing, stale, conflicting, or disallowed; stop and preserve the gap rather than guessing provenance. For input-grounding errors, name the missing or out-of-scope path, journey, claim, or expected result and do not invent a replacement. The manifest, its journeys, and its traces are not repaired to make a result pass.

For command errors, identify the claimed invocation and the allowlist entry it fails to match. Reject a command that is absent from the JSON `command_allowlist` , or whose contract, manifest, test, evidence, or other repository-relative target no longer resolves. For stale-reference errors, show the old or unresolved page ID, path, source ID, acceptance check, or manifest link and stop before execution. These messages should let the authorized producer correct one bounded input or page/registry mapping; they must not require an operator to diagnose the cause from a vague failure or to use a newly generated fact as authority.

Valid Contract Path

First parse the two immutable preflight records with the exact declared inspection commands: `python3 -m json.tool artifacts/preflight/source-authority-seal.json` and `python3 -m json.tool artifacts/preflight/pre-authoring-reader-journeys.json` . Next compare the sealed sources and all journey references with the in-scope repository paths, including the page header and matching registry record. Then check that the product-contract evaluator invocation is compatible with the manifest's allowlist; the declared bounded check is `python3 -m pytest -p no:cacheprovider`

`tests/test_product_contract.py -q` . Do not silently broaden a command, substitute a stale path, or treat an unlisted invocation as reproducible.

When all comparisons resolve, record a clear **PASS** that states authority, inputs, command compatibility, and reference freshness were checked. That pass may be consumed by the next governed workflow step without a human operator reverse-engineering the failure or authoring an explanation by hand. The pass keeps the immutable manifest unchanged, preserves the independent evaluation boundary, and remains candidate-only product evidence. If any comparison fails, record the actionable error and stop before budget-consuming execution or semantic interpretation; a valid contract is the only path to proceed.

Acceptance Checks

1. **AC-1 — Authority validation:** The preflight validates the sealed authority references and reports missing, stale, conflicting, or disallowed authority as an actionable error instead of inferring provenance. It compares the authority-listed source IDs with their registered public-safe metadata and preserves unresolved gaps.
2. **AC-2 — Input grounding:** The preflight grounds every journey, input, and material claim in the declared repository-relative inputs and reports missing or unsupported inputs without inventing replacements. Each required path, journey goal, expected result, contract reference, and page/registry mapping is checked before execution.
3. **AC-3 — Allowlist and stale-reference detection:** The preflight verifies that each claimed evaluator command is compatible with the JSON `command_allowlist` and detects stale or unresolved contract, input, and manifest references before execution. An implausible-looking command or path is not accepted without a current, bounded match.
4. **AC-4 — Valid contract proceeds:** A valid contract receives a clear pass and can proceed to governed execution without manual diagnosis, while the immutable manifest, independent evaluation boundary, and candidate-only boundary remain intact. The pass grants no platform-owned authority or publication effect.

Provenance: registry-listed (no candidate packet)

Registry Source: `content/public/pre-authoring-reader-journey-preflight.md`