DOCUMENTATION PAGE

Contract-Derived Candidate Evidence Preflight Receipt

Contract-Derived Candidate Evidence Preflight Receipt

Purpose

This is a read-only feasibility receipt for the Internal Knowledge workstream. It is written for auto-orch maintainers and stage workers, with Internal Knowledge playbook authors and independent readers able to use the same contract. Before candidate authoring, it inventories the authoritative contracts, required journey coverage, artifact schemas, command boundary, expected outputs, deterministic validators, preventable failures, blockers, and ownership needed to make a bounded feasibility decision.

The receipt records only facts established by the declared inputs and preserved validator evidence. A command declaration is not execution evidence, a product-contract result is not a platform attestation, and a missing artifact is not a passing result. Unresolved or inconsistent prerequisites remain visible. This page does not authorize a product change or any later governed transition.

Canonical Inputs

The sealed authority input is `sources/authority.json`, schema version `1`, revision `2026-08-08`. Its `source_files` set is exactly `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, and it records `employee_may_modify: false`. `registries/sources.json` supplies the matching registered records; each is marked `public-safe`. Their authority classes, locators, revisions, and digests may be compared but may not be rewritten here. The digest-backed references to the separate governing repository do not expose private governance material or grant this product repository control over it.

The contract inputs are `content/candidate-evidence-handoff-preflight.md`, `docs/contract-derived-candidate-evidence-preflight-receipt.md`, `models/evidence-artifact-glossary-contract.md`, `tests/test_product_contract.py`, and `tools/validate_content.py`. Repository guidance comes from `AGENTS.md`, `docs/project-onboarding.md`, and `questions-for-the-human.md`. The smoke boundary is described by `tests/smoke_manifest.agent-orch.json` and `tests/check_documentation_smoke.py`. The authored output set is this page, its journey manifest, and the matching page-registry entry; named paths are not treated as available evidence unless an authoritative record supplies them.

The glossary contract requires authority and reader-evidence artifacts to stay distinct from product conclusions and platform-owned attestations. It also names `artifacts/glossary-authoring/authority-seal.json` and `artifacts/glossary-authoring/evidence-artifact-glossary-journeys.json`, but neither companion artifact is a trusted or declared input for this step. That absence is carried into the feasibility result rather than silently repaired.

Feasibility Receipt

The authority-to-source comparison is feasible as a read-only inspection: all three authority-listed logical IDs are present in the registered source set, and all three registered records are public-safe. This establishes an input inventory, not permission to modify `sources/authority.json`, `registries/sources.json`, external governance references, or the employee boundary. Any later stale, missing, conflicting, or non-public-safe field must be recorded as unresolved.

The synchronized manifest is schema version `1`, kind `reader-journey-manifest`, and authority `mission`. It enumerates exactly AC-1, AC-2, and AC-3. Its required journey is named `Verify the candidate evidence preflight receipt`, is non-exploratory, and traces to every one of those criteria. The journey trace describes what an independent reader may check; it does not claim that this authoring step ran a reader journey.

The manifest command allowlist contains only the non-empty prefix `python3 -m pytest`. Each declared journey uses the exact command `python3 -m pytest tests/test_product_contract.py -p no:cacheprovider` and expects exit code `0`. Its expected observable result is a passing deterministic product-contract validation, preserved separately

from reader execution. The manifest includes no smoke command, shell wrapper, alternate pytest prefix, or claim that an authoring worker executed the reader route.

The preserved validator records supplied for this handoff report successful deterministic checks: semantic-only content validation, JSON syntax validation, the product-contract pytest command, and projection building each have exit status `0` and passed status in their preserved records. For the exact pytest record, validator identity `agent_orch.validators.command_succeeds@1` reports exit status `0`, passed `true`, and evidence hash `f7b97990b624e6b6124e11aa3452aff97720e8d82a216622e752986193ff58f2`. These are supplied platform-owned validator observations, not commands run or self-certified by this authoring step.

Expected authored outputs are the Markdown receipt at `content/contract-derived-candidate-evidence-preflight-receipt.md`, the synchronized reader contract at `journeys/contract-derived-candidate-evidence-preflight-receipt.json`, and one matching entry in `registries/pages.json`. The expected output is synchronization and explicit feasibility status; it is not a release, publication, or reader-execution artifact.

Blockers and Ownership

Blocker B-1 is a contract divergence. The candidate-evidence handoff page describes AC-1 through AC-4 and requires the `python3 -B -m pytest` prefix, while this mission's reader contract preserves AC-1 through AC-3 and the `python3 -m pytest` prefix. The acceptance sets and commands are not interchangeable. The authorized product-contract and reader-manifest maintainer owns reconciliation; the responsible individual is not identified by the declared inputs and remains unresolved.

Blocker B-2 is an artifact-schema gap. The glossary names the two `artifacts/glossary-authoring/` companion files, but their contents, digests, availability, and owner are not supplied in this step. The artifact-contract maintainer who can supply or explicitly withdraw those dependencies owns the gap; no individual identity is established. A path mentioned by prose cannot close this blocker.

Blocker B-3 is missing trusted reader-execution evidence. No trusted artifact input is declared for this handoff, and the authoring role cannot execute or self-certify the reader journey. The platform-owned evaluator must supply a record containing journey identity, exact command, exit status, observed output, source references, acceptance traces, route ownership, and capture location. Until then, the expected pytest result remains an expectation only.

Blocker B-4 preserves the separately recorded environment-specific reader observation: `journey.resolve-reader-journey-harness-decision` used `python3 tests/check_documentation_smoke.py`, exited `1`, and observed `Operation not permitted` because the evaluator environment denied localhost socket access. The follow-up evaluator owns this unresolved evidence. It does not establish a recurring product harness defect, retry count, affected claim, or repair target; those facts remain unresolved and must not be inferred.

The receipt author and stage worker own only the bounded page, manifest, and registry synchronization. They may compare inputs, preserve exact supplied observations, and route gaps, but they cannot waive a contract mismatch, invent a missing owner, close platform evidence, or convert a validator pass into approval. Candidate authoring feasibility therefore remains blocked until the bounded owners supply authoritative reconciliation or missing evidence.

Read-Only Boundary

This page is a product-side feasibility record. It does not claim platform execution, evaluator-route execution, validator authority, machine identity, evidence-chain completion, independent review, human approval, activation, promotion, publication, scheduling, rollback, release-pointer creation, or a live-directory effect. Candidate-only means that a possible handoff remains parked and bounded; it does not mean approved, publishable, or ready for a transition.

Only the three declared product outputs are in scope: this page, `journeys/contract-derived-candidate-evidence-preflight-receipt.json`, and its matching `registries/pages.json` entry. The receipt does not edit authority, source records, the candidate-handoff page, the glossary, smoke inputs, tests, generated projections, releases, publication pointers, schedules, rollback targets, or a live directory. Generated output remains derived and cannot become semantic source through this receipt.

If a later governed evaluator supplies reader evidence, that record must keep the journey ID, natural-language goal, exact allowlisted command, exit status, observed output, source references, acceptance traces, route owner, and any recurrence facts. An environment or permission failure must remain an evidence blocker until its owning authority establishes product impact. This receipt cannot infer a defect, repair target, approval, or transition from an absent capture, an expectation, or a supplied deterministic validator result.

Acceptance Checks

- **AC-1 — Canonical authority and inputs:** The reader identifies authority revision `2026-08-08`, the exact three logical source IDs, matching public-safe registry records, `employee_may_modify: false`, and the declared contract, guidance, smoke, validator, and output inputs. Private governance material, missing provenance, absent artifacts, and unregistered sources remain outside the usable evidence boundary.

- **AC-2 — Coverage and deterministic command:** The reader parses this manifest, confirms exactly AC-1 through AC-3, verifies that required non-exploratory natural-language journeys trace to every criterion, and checks the exact `python3 -m pytest` prefix, exact pytest command, and expected exit code `0`. Preserved validator evidence stays distinct from a reader journey that this authoring step did not claim to execute.

- **AC-3 — Blockers, ownership, and boundary:** The reader records B-1 through B-4, routes each to its bounded owner while retaining unknown identities as unresolved, separates product evidence from platform attestations, and confirms that no approval, activation, promotion, publication, scheduling, rollback, release-pointer, or live-directory action follows from the receipt.

These checks establish only a synchronized, read-only authoring contract. They do not resolve the acceptance divergence, supply the glossary companions, create reader execution evidence, or classify the localhost observation as a product defect. Those outcomes require authoritative follow-up from the owners named above.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/contract-derived-candidate-evidence-preflight-receipt.md`