

DOCUMENTATION PAGE

Candidate Evidence Handoff Preflight and Template

Candidate Evidence Handoff Preflight and Template

Purpose

Workstream: Internal Knowledge.

Audience: Lee and auto-orch operators, including Agent-Orch maintainers preparing an operator-facing evidence handoff, together with future Internal Knowledge playbook authors and independent readers who need to reconstruct the handoff.

Intended outcome: A bounded, auditable packet that lets a maintainer show what was checked, which authority owns each fact, where the result lives, and what remains unresolved without turning a product conclusion into a platform attestation or a transition decision.

Success hypothesis: If acceptance coverage, smoke artifacts, command allowlisting, and output locations are checked before the checklist is drafted, a reader can re-execute the documented validation and distinguish a missing fact from a failed product check.

Validation journey: `journey.candidate-evidence-handoff-documentation-validation` in `journeys/candidate-evidence-handoff-manifest.json`.

The handoff is candidate-only documentation evidence. It does not modify the sealed authority, copy private governance material, establish a recurring product defect from one environment-specific reader failure, or authorize approval, activation, promotion, publication, scheduling, rollback, a release pointer, or a live-directory write. Keep those boundaries visible while the operator assembles the evidence.

Preflight Procedure

Before drafting a checklist, read the sealed preflight receipt at `artifacts/preflight/candidate-evidence-handoff.json` and confirm that the selected workstream is **Internal Knowledge**, the audience is the maintainers and operators named above, and the intended handoff outcome is still bounded. Then verify acceptance coverage: AC-1, AC-2, and AC-3 must each have a concrete artifact, a source reference, an authority owner, and at least one journey trace. A journey that is exploratory may inform discovery, but every non-exploratory journey must trace to one or more of those three checks.

Next inspect the smoke artifacts before writing prose about execution. Confirm that `tests/smoke_manifest.agent-orch.json` names the expected start and check contract, that `tests/check_documentation_smoke.py` resolves its page and registry inputs, and that the documentation-validation journey allowlists the exact re-executable command `python3 -B tests/check_documentation_smoke.py`. The command is a reproducible claim about how to evaluate the documentation; its presence in the manifest is not proof that a platform ran it. Record the observed exit status and output separately, including the current environment-specific `Operation not permitted` reader-gate blocker when that is the result. Do not replace it with a guessed product defect or a successful result from another route.

Finally, verify every command before checklist drafting: it must be a non-empty JSON string in `command_allowlist`, point to the declared repository inputs, and have a named output location. Compare the page metadata with its single entry in `registries/pages.json`, and compare the three authority-listed source IDs with their public-safe records in `registries/sources.json`. Preserve the independent validator evidence supplied by the governing handoff; do not self-certify it or rerun a preserved system-validator result as a substitute. The manifest's `python3 -B` commands are the reader-reexecution allowlist. A preserved record's declared command is copied verbatim as evidence and is not silently changed to the current reader command. Only after

Evidence-Handoff Template

Copy one row for every required artifact and complete every column. “Authority” names the boundary that owns the fact, while “Source reference” identifies the repository-relative record or preserved evidence that a reader can inspect. “Validation result” must state what was actually observed, not what the operator expected. “Command” must be the exact allowlisted invocation used or claimed, and “Output location” must point to the resulting record. If a check was not run, say so. If the evidence is blocked, retain the exact blocker and leave the conclusion unresolved.

Required artifact	Authority	Source reference	Validation result
Sealed authority and source boundary	Platform-owned authority boundary	<code>sources/authority.json</code> ; <code>registries/sources.json</code>	Preflight observed revision <code>2026-08-08</code> , exactly <code>source.naming-decision</code> , <code>source.documentation-boundary</code> , and <code>source.platform-delivery-contract</code> , <code>employee_may_modify: false</code> , and public-safe records; AC-1 preflight passed
Selected workstream and handoff scope	Mission	<code>registries/workstream-selection-receipt.json</code> ; <code>artifacts/preflight/candidate-evidence-handoff.json</code>	Receipt records Internal Knowledge , the named maintainer/operator and independent reader audience, a bounded evidence handoff, and no affected public claim established
Acceptance coverage	Mission	<code>artifacts/preflight/candidate-evidence-handoff.json</code> ; this page; <code>journeys/candidate-evidence-handoff-manifest.json</code>	AC-1, AC-2, and AC-3 each have a concrete artifact and one non-exploratory journey whose <code>acceptance_checks</code> and <code>traces_to</code> sets agree
Smoke manifest and checker	Platform-owned reader-gate boundary	<code>tests/smoke_manifest.agent-orch.json</code> ; <code>tests/check_documentation_smoke.py</code> ; <code>artifacts/preflight/candidate-evidence-handoff.json</code>	One supplied reader attempt for <code>journey.resolve-reader-journey-harness-decision</code> used <code>python3 tests/check_documentation_smoke.py</code> , exited and observed <code>Operation not permitted</code> because localhost socket access was denied
Preserved product-contract validator evidence	Platform-owned validator authority	<code>preserved-validator-evidence.json</code> supplied by the Agent-Orch run	Three preserved records each report exit status <code>0</code> , passed <code>true</code> , <code>10</code> passed, <code>0</code> failed, and <code>0</code> skipped. Hashes, in supplied order, are <code>a0cc3d856871267cc247dab9c472f657558b1e253efe31cc2515a25badda9d</code> ; <code>9347adc57824c6378b15a155a5f3a716e14588d7549d28a5cde2c485c9a69a</code> and <code>63699490ab521d98b12e507f2b11c252e0fd2f6476dd8747ece84cfc6516f</code>
Page, registry, and journey outputs	Author, subject to the product contract	<code>content/public/candidate-evidence-handoff.md</code> ; <code>registries/pages.json</code> ; <code>journeys/candidate-evidence-handoff-manifest.json</code>	Metadata and registry identity agree; the pre-existing required headings are retained and the provenance matrix and decision headings are added; AC-1, AC-2, and AC-3 trace; and the two non-empty allowlisted commands agree

8/16/26, 3:03 PMCandidate Evidence Handoff Preflight and Template — Documentation Portal

For a handoff that is not yet complete, replace placeholders with explicit `unresolved` values and keep the original observation beside any later result. The minimum evidence record includes the artifact path, authority boundary, source reference, command, exit status, observed output, output location, acceptance trace, and blocker owner or `owner unresolved`. The preserved validator results are independent evidence: each supplied record's `10 passed`, `0 failed`, `0 skipped`, exit status `0`, command, and evidence hash must be copied only from that preserved validator artifact. The three records are retained separately because they support different upstream acceptance relationships; none is a new local validator claim. The smoke result is recorded from the permitted reader-gate environment, where the one localhost denial is unresolved. A product page may explain these distinctions, but it cannot certify platform-owned execution.

Candidate-Evidence Contract Provenance Matrix

This matrix maps each material packet claim to the source or evidence boundary that owns it, the deterministic route that can check it, the coverage actually visible in the supplied records, and a classification. A declared command is a route, not proof of execution; a platform-owned attestation is not replaced by an author assertion. “Not rerun” records the boundary of this handoff and does not turn an unobserved result into a pass.

Candidate-evidence claim or contract obligation	Authoritative source or evidence boundary	Existing deterministic test or validation route
Internal Knowledge scope, audience, bounded outcome, success hypothesis, and validation journey	<code>registries/workstream-selection-receipt.json</code> ; <code>journeys/candidate-evidence-handoff-manifest.json</code>	Manifest journey <code>journey.candidate-evidence-handoff-preflight</code> ; <code>declare:cacheprovider tests/test_product_contract.py</code>
Sealed authority revision <code>2026-08-08</code> , exactly three authority-listed source IDs, and <code>employee_may_modify: false</code>	<code>sources/authority.json</code> ; preserved Agent-Orch validator evidence	Preserved <code>PYTHONDONTWRITEBYTECODE=1 python3 -m json.tool sources/</code>
The three authority-listed IDs have public-safe source-registry records with their recorded class, locator, revision, and digest	<code>registries/sources.json</code> joined to <code>sources/authority.json</code>	Existing packet's explicit source-boundary preflight command; <code>python3 -m json.</code> validates the preserved authority record
Internal Knowledge is the selected queue and the exact three-queue comparison remains visible before item impact or feasibility	<code>registries/workstream-selection-receipt.json</code>	Candidate preflight journey declaration; declared <code>python3 -B -m pytest -p no tests/test_product_contract.py</code> route
Candidate packet acceptance coverage is exactly AC-1 through AC-3, with one required non-exploratory journey and matching <code>acceptance_checks</code> and <code>traces_to</code> for each	<code>journeys/candidate-evidence-handoff-manifest.json</code>	<code>tests/test_candidate_evidence_preflight.py::test_manifest_match::test_manifest_model_and_fixture_share_journey_declarations</code>
The candidate handoff allowlist and each journey command are exact, non-empty strings	Candidate handoff manifest <code>command_allowlist</code> and journey declarations	<code>tests/test_candidate_evidence_preflight.py::test_allowlist_is_exact::test_command_claims_are_not_normalized_before_allowlist_match::test_allowlisted_command_must_match_the_declared_journey_command</code>
Evidence records retain exact names, traces, steps, source references, command claims, exit codes, observed results, and the independent review-verdict requirement	<code>docs/candidate-evidence-preflight-contract.md</code> ; candidate manifest; platform-owned review boundary <code>code-reviews/content-review.verdict.json</code>	<code>tests/test_candidate_evidence_preflight.py</code> fixture, schema, CLI, and

Candidate-evidence claim or contract obligation	Authoritative source or evidence boundary	Existing deterministic test or validation route
The documentation smoke route and its supplied reader result remain distinct	<code>tests/smoke_manifest.agent-orch.json</code> ; <code>tests/check_documentation_smoke.py</code> ; platform-owned reader-gate evidence	Candidate manifest route <code>python3 -B tests/check_documentation_smoke.py</code>
Existing packet text retains three platform-owned product-contract validator records, each reported as exit <code>0</code> , <code>10</code> passed, <code>0</code> failed, and <code>0</code> skipped, with distinct hashes	<code>preserved-validator-evidence.json</code> supplied by the Agent-Orch handoff; validator authority remains platform-owned	Existing packet's copied <code>PYTHONDONTWRITEBYTECODE=1 python3 -m pytest tests/test_product_contract.py</code> claims
Page metadata, page registry identity, journey references, and candidate-only transition boundaries remain coherent	This page; <code>registries/pages.json</code> ; candidate manifest; receipt <code>forbidden_actions</code>	<code>python3 tools/validate_content.py</code> ; <code>tests/test_product_contract</code> release tests

Mismatch Classification and Lee Decision

The matrix separates four states. Confirmed alignment means the source declarations agree, not that an unrun command passed. Evidence gaps identify a missing artifact or an authority boundary not covered by the preserved result. Source/test mismatches identify declarations that do not line up with the test route currently named. Genuinely unresolved choices are limited to the one operator decision below; they do not authorize a transition.

Confirmed alignment. The preserved authority command is successful, the authority JSON retains revision `2026-08-08` , the three listed source IDs, and `employee_may_modify: false` , and the workstream receipt and candidate manifest keep the Internal Knowledge, candidate-only scope visible. The packet's provenance claims therefore remain bounded to the sources and evidence records named above.

Evidence gaps. The preserved result does not test the authority-to-source registry join, the page-to-registry join, the candidate page's review-verdict artifact, or the prior product-contract records. The named review path is a requirement, not proof that `code-reviews/content-review.verdict.json` exists or is clean. Those gaps keep handoff completion unresolved.

Source/test mismatches. The candidate handoff manifest declares three candidate journeys and two `python3 -B` commands, while the current `journeys/user_journeys_manifest.json` is a different publication-gap manifest with different journey names and four JSON commands; the candidate test file explicitly compares those manifests. The general preflight contract also describes its own synchronized five-check route. Separately, the supplied smoke attempt omitted `-B` relative to the candidate allowlist. These are declaration and evidence-routing mismatches to reconcile at the owning source boundary, not evidence of a product harness defect.

Genuinely unresolved choice for Lee. Retain this as a candidate-only, environment-specific evaluation blocker and route one permitted follow-up in an environment with localhost socket access. Do not name a recurring product harness defect, affected claim, or repair target from the single `Operation not`

`permitted` attempt. No further decision is needed here about approval, promotion, publication, scheduling, rollback, a release pointer, or a live-directory write; those controls remain outside this packet.

Acceptance Checks

- **AC-1 — Acceptance coverage and authority:** The reader verifies the **Internal Knowledge** workstream, audience, intended outcome, success hypothesis, validation journey, sealed authority revision `2026-08-08`, exactly three authority-listed public-safe source IDs, and `employee_may_modify: false`. The reader also confirms that AC-1, AC-2, and AC-3 each have a non-exploratory natural-language journey trace before checklist drafting, without copying private governance material or inventing authority.
- **AC-2 — Evidence inventory and ownership:** The reader records every required artifact with its owning authority, source reference, observed validation result, exact command, output location, and unresolved blocker. The inventory includes the smoke manifest and checker, all three preserved validator records with their distinct hashes, and the page/registry/journey outputs. Product conclusions remain separate from platform-owned attestations; the localhost-denied `Operation not permitted` observation is an environment-specific unresolved blocker, not evidence of recurrence, an affected claim, or a repair target.
- **AC-3 — Re-executable documentation validation:** The documentation- validation journey contains the non-empty allowlisted command `python3 -B tests/check_documentation_smoke.py`. The reader can locate the supplied result for `journey.resolve-reader-journey-harness-decision`, its claimed command `python3 tests/check_documentation_smoke.py`, exit status `1`, and `Operation not permitted` output, then route a permitted follow-up in an environment with localhost socket access. The handoff stays candidate-only and performs no approval, activation, promotion, publication, scheduling, rollback, pointer change, or live-directory write.

Provenance: registry-listed (no candidate packet)

Registry Source: `content/public/candidate-evidence-handoff.md`