DOCUMENTATION PAGE

Evidence Handoff Preflight and Template

Evidence Handoff Preflight and Template

Purpose

The primary workstream for this preflight is **Internal Knowledge**. Its audience is agent-orch maintainers preparing an operator-facing candidate evidence handoff, Internal Knowledge playbook authors who will reuse the format, and independent readers who need to inspect the result. The intended outcome is a bounded, readable handoff that explains the reader problem, recommendation, acceptance coverage, evidence ownership, and unresolved facts well enough for a later checklist author to work from the same inputs. This is product-side documentation guidance: it does not change sealed authority, create a platform attestation, or authorize approval, activation, promotion, publication, scheduling, rollback, a release pointer, or a live directory.

The preflight is deliberately earlier than checklist authoring. It turns the proposed handoff into an evidence inventory before prose can make an unsupported conclusion look complete. A maintainer should be able to say which workstream was selected, what outcome the reader needs, which sources own each fact, what smoke and journey records exist, which deterministic commands can be re-executed, and which version or rollback facts remain unresolved. Missing evidence stays visible instead of being filled with plausible wording.

Preflight Procedure

Begin by recording the selected workstream, audience, reader problem, and intended outcome. Then compare the proposed acceptance criteria with the reader-journey manifest: every criterion must have a

concrete artifact, an owning authority or explicitly unresolved owner, and a non-exploratory journey trace. If acceptance coverage is missing, name the missing criterion and stop before drafting a checklist; a checklist cannot repair absent evidence. Keep product conclusions separate from platform-owned validator, route, identity, terminal-state, approval, or publication attestations.

Next inspect the smoke artifacts. Confirm that the named smoke manifest, checker, input page, registry record, and captured output location exist or are explicitly marked unresolved. Record the exact exit status and observed output for each supplied attempt, including a permission or environment blocker when that is what the reader observed. A missing smoke artifact is a preflight finding, not a reason to infer that the smoke passed or that a product defect exists. Preserve failed evidence beside any later result.

Finally, check re-executability before checklist authoring. Every deterministic command must be a non-empty string in the selected manifest's `command_allowlist`, resolve to the declared repository inputs, and identify where its output will be recorded. Reject a stale, substituted, or unallowlisted invocation and record the mismatch. Compare the page metadata with its single `registries/pages.json` entry and treat `build/index.json` as generated output. Only when acceptance coverage, smoke artifacts, and exact commands are accounted for should the maintainer draft a checklist and assign a review verdict. The handoff remains candidate-only throughout.

Candidate Evidence Handoff Template

Copy the following template into a candidate record and replace every placeholder with an observed value or the explicit word `unresolved`. Keep the labels stable so an independent reader can compare handoffs. “Authority/source evidence” identifies who owns a fact and where its supporting record lives; it is not a claim that a platform gate ran. “Smoke evidence” and “user-journey evidence” retain exact observations, including failures. “Previous-version identity” and “rollback-target facts” must be copied from the candidate evidence when present and must remain unresolved when no authoritative record establishes them.

```
candidate_evidence_handoff:
  selected_workstream: "Internal Knowledge"
  audience: "<operator-facing maintainers, playbook authors, and/or independe
  intended_outcome: "<bounded reader outcome>"
  reader_problem: "<specific problem the reader needs this handoff to solve>"
  recommendation: "<bounded recommendation, or unresolved>"
  acceptance_criteria:
    - id: AC-1
      statement: "<criterion and its concrete artifact>"
      evidence_location: "<path or stable evidence id>"
      journey_refs: ["<journey id>"]
    - id: AC-2
      statement: "<criterion and its concrete artifact>"
      evidence_location: "<path or stable evidence id>"
      journey_refs: ["<journey id>"]
    - id: AC-3
      statement: "<criterion and its concrete artifact>"
      evidence_location: "<path or stable evidence id>"
      journey_refs: ["<journey id>"]
  authority_source_evidence:
    authority_owner: "<authority boundary or unresolved>"
    source_refs: ["<registered source or evidence path>"]
    observed_result: "<revision, digest, disclosure, and conclusion>"
  smoke_evidence:
    manifest: "<smoke manifest path>"
    checker: "<checker path>"
    command: "<exact observed or allowlisted command>"
    exit_status: "<integer or unresolved>"
    observed_output: "<verbatim-short observation or unresolved>"
    output_location: "<captured result path or unresolved>"
  user_journey_evidence:
    manifest: "<reader-journey manifest path>"
    journeys: ["<journey id>"]
    traces_to: ["AC-1", "AC-2", "AC-3"]
    observed_result: "<reader conclusion, including retained blockers>"
  deterministic_commands:
    - command: "<exact non-empty command from command_allowlist>"
      purpose: "<bounded check>"
```

```
      output_location: "<result path or unresolved>"
      exit_status: "<integer or unresolved>"
  review_verdict:
    value: "<pass, findings, blocked, or unresolved>"
    reviewer_evidence: "<independent review record or unresolved>"
  candidate_only_publication_boundary: "<state that this is candidate-only an
  previous_version_identity: "<exact prior version identifier, or unresolved>
  rollback_target_facts:
    target_identity: "<exact rollback target, or unresolved>"
    available: "<true, false, or unresolved>"
    evidence_location: "<candidate artifact path or unresolved>"
```

After copying, check that every placeholder is resolved or intentionally marked unresolved, each command is exact, and each acceptance criterion points to evidence and a journey. A review verdict describes the product evidence; it does not establish platform execution or human approval. Leave the candidate parked if source authority, smoke output, journey coverage, prior-version identity, rollback target, or any other required fact is missing. The template is complete only when a reader can tell what was observed, what remains open, and who owns the next evidence-producing action.

Acceptance Checks

- **AC-1 — Scope and coverage:** The reader confirms that **Internal Knowledge** is the selected workstream, the named audience and intended outcome are explicit, and the reader problem and recommendation are bounded. Before checklist authoring, the reader maps AC-1, AC-2, and AC-3 to concrete artifacts and non-exploratory journey traces; missing coverage is reported as an unresolved preflight finding rather than hidden in checklist prose.

- **AC-2 — Evidence inventory:** The reader verifies that the handoff records authority/source evidence, smoke evidence, and user-journey evidence with source locations, observed results, ownership, and blockers. Missing smoke artifacts, failed attempts, or platform-owned attestations are retained as facts about the evidence boundary; they are not rewritten as a passing product conclusion. The reader can complete the template without inventing provenance.

- **AC-3 — Re-execution and boundary:** The reader verifies that each deterministic command is exact, non-empty, and present in the selected manifest's `command_allowlist`, with an output

location and observed status. The review verdict, candidate-only publication boundary, previous-version identity, and rollback-target facts are recorded separately, with absent facts marked unresolved. The journey ends without inferring approval, promotion, publication, scheduling, rollback, a pointer, or a live directory.

The synchronized reader manifest maps one non-exploratory journey to each check: the maintainer verifies scope and coverage, the playbook author fills the evidence inventory, and the independent reader verifies command re-execution and the candidate-only boundary. These traces make the preflight auditable before checklist authoring while keeping the handoff within the declared product paths.

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