

Candidate-Only Reader Evidence Preflight Guide

Candidate-Only Reader Evidence Preflight Guide

Purpose and Scope

The primary workstream is **Internal Knowledge**. This guide is for auto-orch operators preparing governed-run evidence, Internal Knowledge playbook authors, and independent readers. Its outcome is a candidate-only reader-evidence packet that a later evaluator can inspect and reproduce: the operator can name the reader problem, intended outcome, exact journey, source or authority, artifact location, acceptance trace, and unresolved gaps without turning documentation into a platform attestation. The guide covers the preflight before or during a governed run; it does not alter sealed authority, the source registry, or an evaluator's independent evidence.

Use the guide to prepare evidence for a bounded reader check, not to make a transition decision. A command declaration, page entry, expected result, or product conclusion is useful input, but none is proof that a command ran or that a gate passed. Keep missing facts explicitly unresolved and keep product evidence separate from platform-owned validator, identity, route, approval, activation, or publication evidence.

Reader-Evidence Preflight

Before checklist authoring, write down the selected **Internal Knowledge** workstream, the operator audience, the reader problem, and the intended outcome. Open the synchronized page header, page registry entry, source authority, source registry, and journey manifest. Confirm that the page has no claim references and uses only the three approved public-safe source refs. Then enumerate AC-1 through AC-4 and match each criterion to one required non-exploratory journey. A criterion without a journey, authority, or artifact location is an unresolved coverage gap, not an invitation to infer completion.

For each journey, prepare one evidence row and reserve space for the exact journey name, the eligible non-empty command prefix, observed exit code, reproducible output, source or authority, artifact location, acceptance trace, and failed-evidence record. The allowlisted command in this guide is `python3 -B -m pytest -p no:cacheprovider tests/test_product_contract.py -q`. It is a declaration to be used by an authorized reader; it is not an observed execution. Replace `unresolved` only with evidence captured by that reader.

Command-Claim Checklist

Complete exactly one row for every required journey in `journeys/candidate-only-reader-evidence-preflight.json`. The status must be `completed` only when the reader record contains the requested observation; otherwise use `unresolved` and name what is missing. The exact allowlisted invocation is both the eligible non-empty command prefix and the command claim for each row. Do not report a command as run merely because it appears here or in the manifest.

Status (completed or unresolved)	Exact journey name	Eligible non-empty command prefix / exact command claim	Observed exit code and reproducible output	Source or authority	Artifact location	Ac tra
unresolved	As an auto-orch operator, confirm the Internal Knowledge scope and intended reader outcome before a governed run.	<code>python3 -B -m pytest -p no:cacheprovider tests/test_product_contract.py -q</code>	unresolved — capture the exit code and output from the governed reader record; a declaration is not execution proof.	<code>sources/authority.json</code> , <code>registries/sources.json</code> ; mission/author journey authority	unresolved — record the governed-run evidence path	AC
unresolved	As an Internal Knowledge playbook author, inventory each reader-evidence field and its owning source or authority.	<code>python3 -B -m pytest -p no:cacheprovider tests/test_product_contract.py -q</code>	unresolved — capture reproducible output and distinguish missing evidence from a passing result.	This guide, <code>registries/pages.json</code> ; author journey authority	unresolved — record the completed inventory path	AC
unresolved	As an independent reader, verify the exact command claim and acceptance trace without inferring execution.	<code>python3 -B -m pytest -p no:cacheprovider tests/test_product_contract.py -q</code>	unresolved — capture the observed exit code and reproducible output; do not substitute an expected result.	<code>tests/test_product_contract.py</code> , this guide; mission journey authority	unresolved — record the independent reader record path	AC
unresolved	As an auto-orch operator, confirm that the evidence packet remains candidate-only and	<code>python3 -B -m pytest -p no:cacheprovider tests/test_product_contract.py -q</code>	unresolved — capture the bounded reader result without treating it as approval or a transition attestation.	<code>sources/authority.json</code> , this guide, <code>registries/pages.json</code> ; mission journey authority	unresolved — record the candidate-only evidence path	AC

Status (completed or unresolved)	Exact journey name	Eligible non-empty command prefix / exact command claim	Observed exit code and reproducible output	Source or authority	Artifact location	Acceptance trace
	preserves every unresolved boundary fact.					

The checklist records evidence claims, not authority to act. It must not infer approval, promotion, publication, scheduling, rollback, a release pointer, activation, or a live-directory action from a command, an exit code, a reader verdict, or a candidate record. If the output is absent, keep the row unresolved; if execution failed, keep the failure and its artifact location available for independent review.

Candidate-Only Boundary

Candidate-only means that the guide and its reader evidence describe a reversible product-side handoff awaiting separately governed decisions. It is not approval, promotion, publication, scheduling, rollback, a release pointer, activation, or permission to write a live directory. The manifest's journeys can establish what a reader should inspect and what product evidence was observed, but they cannot create platform-owned attestations or change the governing repository's authority contract.

Keep the candidate disposition explicit in every operator summary. Separate the product conclusion from platform evidence about execution, evaluator route, identity, gate state, or human authorization. If previous-version identity, rollback-target facts, or a transition record is not supplied by its owner, write `unresolved` . A synchronized page registry proves metadata alignment only; it does not establish a release pointer or a live-directory change.

Failed-Evidence Preservation

A failed command, denied socket, missing artifact, nonzero exit code, or environment-specific observation is evidence about what was observed. Preserve the exact journey name, exact command, exit code, reproducible output, source or authority, artifact location, time or run identifier when available, and acceptance trace. Keep the failed record beside any later retry or successful result so an independent reader can distinguish attempts. Do not delete, rewrite, or silently replace failed evidence with a declaration that the command was eligible.

The operator may mark a row `unresolved` when the result or owner is missing. That label is honest evidence status, not a product defect and not a reason to invent a repair target. A preserved failure also does not prove recurring harness failure, approval, promotion, publication, scheduling, rollback, activation, or a live-directory action. Route a permitted follow-up evaluation to the authority that owns the missing fact while retaining this candidate-only boundary.

Acceptance Checks

- **AC-1 — Scope and outcome:** The reader identifies Internal Knowledge as the primary workstream, names the operator audience, states the reader problem and intended outcome, and traces that coverage to the required operator journey without claiming that a command ran.
- **AC-2 — Evidence inventory:** The reader can complete or mark unresolved every checklist field: exact journey name, eligible non-empty command prefix, observed exit code and reproducible output, source or authority, artifact location, acceptance trace, and preservation of failed evidence.
- **AC-3 — Command claim boundary:** The reader verifies that every journey uses the exact allowlisted command and distinguishes that declaration from observed execution. Missing output, a nonzero exit code, or a failed attempt remains visible and does not become an approval or product conclusion.
- **AC-4 — Candidate-only handoff:** The reader confirms that the packet is candidate-only and does not infer approval, promotion, publication, scheduling, rollback, a release pointer, activation, or a live-directory action. Product conclusions and platform-owned attestations remain separate.

Acceptance is complete only when each required journey has one completed or unresolved row and every unresolved or failed fact is named for follow-up. The guide prepares evidence for a governed reader; it does not perform the reader run or authorize any later transition.

Provenance: registry-listed (no candidate packet)

Registry Source: `content/candidate-only-reader-evidence-preflight-guide.md`