DOCUMENTATION PAGE

Authoritative Reader-Journey Contract Compatibility Checks

Authoritative Reader-Journey Contract Compatibility Checks

Purpose

This page gives agent-orch maintainers and governed-run operators a repeatable compatibility check for a reader-journey manifest, its page metadata, and its evidence claims. It keeps the result candidate-only: the procedure can show a product mismatch or a reproducible product pass, but it cannot change sealed authority, grant validator authority, approve a run, publish content, or create a release pointer. Read the exact manifest names and goals as the contract a later independent reader must be able to replay.

Authoritative Contract

The sealed authority is the source boundary. `sources/authority.json` has revision `2026-08-08`, names exactly `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, and sets `employee_may_modify` to `false`; the matching records in `registries/sources.json` must remain public-safe. The compatibility manifest is authoritative for its own enumerated AC-1 through AC-4 criteria, exact journey `name` strings, natural-language `goal` text, `authority` values, `command_allowlist` strings, and `traces_to` mappings. A journey authority is permitted only when it is exactly `human`, `mission`, `author`, or `exploratory`; a required non-exploratory journey uses `human`, `mission`, or `author`, and an exploratory journey cannot replace required coverage.

Exact journey names are stable evidence keys, not display suggestions. A `commands_run` claim records the exact allowlisted command, exit status, and observed result for the manifest journey whose name it

carries. The natural- language goal explains what an independent reader starts with and can observe; it does not authorize extra commands. A command claim is valid only when its string matches one `command_allowlist` entry byte-for-byte. Product evidence from that claim remains separate from platform-owned attestations.

Compatibility Checks

Run this procedure from the repository root before accepting a compatibility result:

1. Open `journeys/reader-journey-contract-compatibility.json` and enumerate AC-1, AC-2, AC-3, and AC-4. For every required non-exploratory journey, compare `traces_to` with the declared AC IDs and compare the full `name` and natural-language `goal` with the journey being evaluated. If any AC has no trace, or a journey traces to an undeclared AC, record a missing-AC-trace mismatch and stop; do not invent a trace or silently drop the journey.
2. Validate each journey `authority` as an exact enum string. A value such as `Mission`, `operator`, `human`, or a missing value is malformed even if a reader could guess its intent. Required non-exploratory coverage must not be converted to exploratory coverage to hide the malformed value.
3. For each `commands_run` claim, compare the command string byte-for-byte with the manifest `command_allowlist`. A command that is merely similar, reordered, or wrapped in an unlisted shell command is not permitted. Record the offending command, the journey name, and the allowlist comparison as a product mismatch; do not execute an unpermitted command to obtain a pass.
4. Key every result by the exact manifest journey `name`, including case, punctuation, and spacing. A failed result keyed by a journey ID, filename, shortened label, translated title, or any other string is not compatible; preserve it as a failed or mis-keyed record and request repair before interpreting its exit status or output.

The four checks are independent gates: a passing command cannot compensate for a missing AC trace, malformed authority, or mis-keyed result. Keep the exact failure, command, exit status, observed output, and comparison path available to the next reviewer so the procedure distinguishes a product conclusion from an unavailable platform execution.

Repair and Re-execution

Repair only the bounded product mismatch named by the comparison. Page prose belongs in `content/reader-journey-contract-compatibility.md`, synchronized page identity and metadata belong in its one object in `registries/pages.json`, and manifest names, goals, authorities, traces, or command strings belong in `journeys/reader-journey-contract-compatibility.json`. Do not modify `sources/authority.json` or `registries/sources.json` to make a reference pass, and do not hand-edit generated output, releases, publication pointers, or a live directory. A repaired field must still be checked against the sealed source boundary and the page header/registry pair.

After repair, an independent reader reopens the manifest, repeats the four compatibility checks, and re-executes only an exact allowlisted command from the journey under review. The resulting `commands_run` claim must carry the same exact journey name, command string, exit status, and observed result. Retain the earlier failed or mis-keyed record beside the later result rather than overwriting history. A successful local product command establishes only the observed product result; it does not attest route selection, validator authority, approval, activation, promotion, publication, scheduling, rollback, or a release pointer.

Acceptance Checks

- **AC-1 — sealed source boundary:** The reader verifies the authority revision, exact three authority-listed source IDs, and `employee_may_modify: false`, then confirms that each matching registry record is public-safe without copying private governance material or editing the protected source files.
- **AC-2 — journey compatibility:** The reader enumerates the exact manifest journey names, confirms each natural-language goal and required `traces_to` mapping covers one or more of AC-1 through AC-4, and rejects a missing trace, duplicate identity, malformed authority enum, or exploratory substitute for required non-exploratory coverage.
- **AC-3 — command evidence:** The reader compares every `commands_run` command claim byte-for-byte with `command_allowlist`, including `python3 -B tools/validate_content.py`, and rejects a command not permitted by that list while retaining its exact exit status and observed result as a product-side compatibility finding.

- **AC-4 — independent keyed re-execution:** An independent reader re-runs an exact allowlisted command and records the result under the exact manifest journey name. A failed result keyed by anything else is incompatible, and a later pass does not erase the earlier failure or expand the candidate-only boundary.

Workstream Selection

This page is the **Internal Knowledge** workstream for agent-orch maintainers and governed-run operators. The sealed selection record makes Internal Knowledge the reproducible choice because it is eligible, underserved in scheduling tier 1, and oldest among the eligible underserved queues before any item-level impact or feasibility score is considered; its selected candidate is the Failure-Triage Decision Guide. **Employee Onboarding** was not selected because it is eligible but not marked underserved, so the ordered rule excludes it before item-level scoring. **Customer Education** was likewise eligible but not underserved and was excluded at the same stage. Both remain visible in the comparison and are not permanently excluded. This workstream choice scopes the documentation route only; it does not schedule work or authorize a transition.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/reader-journey-contract-compatibility.md`