DOCUMENTATION PAGE

Start Here

Start Here

This page is the bounded entry path for the Employee Onboarding workstream. It serves future human engineers and AI employees who arrive with no repository context. Its scope is deliberately narrow: a reader can locate approved public-safe source metadata, orient to the product-owned system, and complete one bounded semantic check. This is a documented onboarding route, not a claim that productivity, approval, readiness, or platform authority has been measured. Begin with `content/public/engine-overview.md`, read these sections in order, and keep the three approved source identifiers visible. Before the check, read `docs/employee-onboarding-first-task-contract.md` and `journeys/employee-onboarding-first-task-manifest.json`; the manifest records the journey goals and command allowlist.

Find Authority

Open `sources/authority.json` first. Its `source_files` list identifies the three approved records: `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`; its `employee_may_modify` field is `false`. Then open `registries/sources.json` and match each ID to its public-safe record: the naming decision is a `canonical-platform-source` at `factory:decisions.md#Governed Execution Documentation Steward naming and route`, the documentation boundary is a `product-owned-public-safe-source` at `product:architecture.md#Semantic source`, and the platform delivery contract is a `canonical-platform-source` at `agent-orch:docs/delivery-profile-contract.md#content`. Confirm each record's revision, digest, and `public-safe` disclosure. The external governance references in the authority file are pointers, not authority granted to this page. If a statement is not represented by the sealed IDs and registry records, leave it unresolved rather than adding a source or filling the gap with plausible prose.

Understand the System

Use `architecture.md` for the repository map. `content/` is the semantic source of constrained Markdown; `registries/` records claims, pages, sources, visuals, and protected identity records; and `models/` holds canonical architecture or diagram data. `tools/build_projection.py` deterministically derives `build/index.json` from semantic source, so generated output cannot replace the source. The local release boundary is separate from onboarding: `releases/` and the local `publication/current-release.json` pointer are not touched by this first task, and the repository has no live-directory writer. For orientation, read `models/engine-overview.json` and follow its declared flow from Semantic source to Evidence model to Reading modes. Use that model to understand the product structure, not to infer an unrecorded claim or any external governance decision.

Synchronize a New Content Page

Use this bounded procedure when a future human engineer or AI employee adds a semantic Markdown page.

1. Read `sources/authority.json` and `registries/sources.json`; choose the intended semantic path and page title, and use only source IDs that are registered there with `public-safe` disclosure.
2. Author the page and assign one stable page ID, keeping its metadata block explicit for `visibility`, `audiences`, `reading_modes`, `claims`, and `source_refs`.
3. In the same change, copy the page title, exact path, stable page ID, visibility, audiences, reading modes, claim refs, and source refs into the matching `registries/pages.json` entry. Every claim and source must resolve, and a public page must remain within the public-safe source boundary. This onboarding page is already registered as `page.employee-onboarding-first-task`; preserve that page ID and its existing page-registry entry rather than creating a second page.
4. Run `python3 tools/build_projection.py`, then inspect generated `build/index.json` for the matching page ID, path, title, visibility, claim refs, source refs, content revision, and authority identity. A page is incomplete until its metadata and projection agree.
5. Run the documented deterministic validator, `python3 tools/validate_content.py`, and this route's bounded command, `python3 tools/validate_content.py --journey journey.employee-onboarding-first-task --semantic-only`; retain the observed results as product evidence.

`build/index.json` is generated output, not a hand-edited source. Missing authority or unresolved references stop the procedure; publication, promotion, rollback, and live-directory actions remain out of scope.

Complete Your First Task

From the repository root, first confirm that the contract and pinned journey manifest describe this route, then run the existing bounded content check listed for the journey:

```
python3 tools/validate_content.py --journey journey.employee-onboarding-first
```

Retain the exact command and its observed result as product-side first-task evidence. The check exercises the deterministic content validator for this onboarding journey; it does not attest platform authority, employee activation, approval, publication, promotion, scheduling, release-pointer state, or any governing control. If it reports a gap, record the exact gap and keep any repair within the assigned content path. A passing result is evidence about this product documentation route only.

Evidence and Boundaries

Trace onboarding guidance only to `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, as sealed in `sources/authority.json` and registered in `registries/sources.json`. Do not invent product, security, pricing, customer-outcome, or operational claims, and do not copy private governance material or runtime evidence. The separate governing repository owns the employee charter, authority contract, routing mandate, validator authority, activation rules, and publication policy. Product conclusions from this page or its validator remain distinct from those platform-owned controls. Generated projection output is not a source replacement, and this first task does not create a release pointer, promote or publish anything, or write a live directory.

Acceptance Checks

- **AC-EO-1** — Locate `sources/authority.json` and `registries/sources.json`, identify all three authority-listed IDs, and match each to its public-safe registry record, locator, revision, and digest without inventing authority or copying external governance material.
- **AC-EO-2** — Use `architecture.md` to explain the roles of `content/`, `registries/`, and `models/`, distinguish semantic source from deterministic projection, and keep the local release boundary separate from the model flow from Semantic source to Evidence model to Reading modes.
- **AC-EO-3** — From the repository root, read the onboarding contract and journey manifest, complete the exact bounded command in this journey, and retain its product-side result as evidence distinct from platform-owned authority, approval, publication, promotion, scheduling, and release-pointer decisions.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/employee-onboarding-first-task.md`