x_make_yahw_x â€” Intake Canary Manual

This canary is the first executable every new environment touches. A simple greeting, yes, but it proves logging, packaging, and orchestrator signaling are wired before we escalate.

Mission Log

- Verify interpreter resolution and module packaging on a clean workspace.
- Exercise shared logging and exporter hooks so `make_all_summary.json` records the Intake Reconnaissance column correctly.
- Provide the visitor with a low-risk target to confirm compliance pipelines are awake.
- Set the baseline standard for documentation and QA across the lab.

Instrumentation

- Python 3.11 or newer.
- Ruff, Black, MyPy, Pyright, pytest if you run the QA net.

Operating Procedure

1. `python -m venv .venv`
2. `\.venv\Scripts\Activate.ps1`
3. `python -m pip install --upgrade pip`
4. `pip install -r requirements.txt`
5. `python -m x_make_yahw_x`

The run emits the greeting and registers evidence in the orchestrator summary. Capture the JSON artefact alongside Change Control notes.

Latest Evidence Snapshot â€” 2025-11-09

- Executed via Environment Studio plan `yahw_smoke` inside warehouse environment `x_venv_make_yahw_x`.
- Run id `f99ea45499a04b44b6f4dc3f64ba342b` completed successfully in 6.266 seconds (Python 3.12; base image `mcr.microsoft.com/devcontainers/python:3.12`).
- Output payload: `{"message": "Hello world!", "schema_version": "x_make_yahw_x.run/1.0", "status": "success"}` with context key `plan` recorded.
- Evidence paths: run directory `c:/x_all_venv_x/runs/20251109_202817_f99ea45499a04b44b6f4dc3f64ba342b`, history entry `c:/x_all_venv_x/history/x_venv_make_yahw_x/20251109T202825Z_f99ea45499a04b44b6f4dc3f64ba342b.json`, plan bundle `c:/x_all_venv_x/studio_plans/20251109T202817Z-a7523a`.

Evidence Checks

| Check | Command | | --- | --- | | Formatting sweep | `python -m black .` | | Lint interrogation | `python -m ruff check .` | | Type audit | `python -m mypy .` | | Static contract scan | `python -m pyright` | | Functional verification | `pytest` |

Reconstitution Drill

During the monthly rebuild I run this canary first: recreate the environment, execute the greeting, confirm the orchestrator logs the intake evidence, and record runtime plus interpreter version. Any surprise leads to immediate documentation and Change Control updates.

Conduct Code

Keep the sample tight. Any message change or behaviour shift must be justified in Change Control with verification steps. The baseline only moves when the evidence demands it.

Sole Architect's Note

I wrote this canary to embody the labâ€™s discipline. Logging, packaging, orchestrator signalingâ€”every line comes from the same hand so accountability stays singular.

Legacy Staffing Estimate

- Without AI acceleration youâ€™d still need: 1 automation engineer, 1 QA lead, 1 documentation specialist.
- Timeline: 4â€“5 engineer-weeks to reach this level of governance.

- Budget: USD 30kâ€“45k to match the embedded evidence standards.

Technical Footprint

- Language: Python 3.11+ with minimal dependencies.
- Tooling: Shared logging and subprocess utilities from `x_make_common_x`, QA stack (Ruff, Black, MyPy, Pyright, pytest).
- Integration: Orchestrator intake stage, visitor verification, Change Control evidence captured during monthly drills.