# **Lean pre-coding sheet — 72/89**

**Component:** fixtures/annex\_b/part\_0/manifest.json (Part 0 **run manifest** fixture)  
 **Version/FormulaID:** This is **data**; FID covers rule primitives, not per-run inputs.

## **1) Goal & success**

**Goal:** Provide a **complete, unambiguous manifest** that pins engine/formula, RNG mode/seed, canonicalization policy, and the exact input artifacts (with SHA-256) for a reproducible run.

**Success:** Schema passes; exactly **one** Registry and **one** ParameterSet; **exactly one of** Ballots **or** BallotTally; seed decodes to **32 bytes**; canonicalization tag matches constant; IO can verify file hashes and the pipeline can lock seed and compute hashes.

## **2) Scope**

**In scope:** engine{version,formula\_id,build?}, created\_utc, rng{mode,seed}, **canonicalization** tag, inputs[] {kind, sha256, length?, path?, id?}, optional meta.

**Out of scope:** Recomputing hashes (done by IO), enforcing JSON canonicalization at write-time (done by IO), executing the run.

## **3) Inputs → outputs**

**Input artifact:** manifest.json (validated by schemas/manifest.schema.json).

**Output to system:** typed **Manifest** → IO verifies file hashes; pipeline locks RNG and contributes to Result/RunRecord hashing (canonical JSON, LF, sorted keys, UTC).

## **4) Entities/Tables (minimal)**

## **5) Variables (only ones used here)**

**None.** Parameterization lives in **ParameterSet**; seed value lives here but is **not** a VM-VAR.

## **6) Functions (signatures only)**

*(Fixture only; no functions.)*

Validation invariants used by the schema/loader (for reference): require\_exactly\_one(DivisionRegistry), require\_exactly\_one(ParameterSet), require\_exactly\_one\_of(Ballots|BallotTally), validate\_seed\_hex\_len\_32(), validate\_canonicalization\_tag(), validate\_sha256\_format\_all().

## **7) Algorithm outline (how it’s consumed)**

Parse **manifest.json**.

Validate engine fields, RNG **mode ∈ {order,rng}** and **seed = 32-byte hex**.

Require canonicalization tag to equal the agreed constant.

Enforce **exactly one** Registry and **exactly one** ParameterSet; **exactly one of** Ballots | BallotTally.

For each input: sha256 = 64 lowercase hex; nonnegative length? if present.

State flow: **load → schema-validate (manifest) → file-hash verify (IO) → lock seed → run pipeline**.

## **8) State flow (very short)**

Used **before** VM-FUN-001 loads artifacts, to ensure a reproducible selection; IDs and seeds echo later in **RunRecord**.

## **9) Determinism & numeric rules**

Canonicalization policy must be the fixed JSON form (UTF-8, **sorted keys**, single trailing \n; **UTC** timestamps). Hashing uses **SHA-256** over canonical bytes.

Seed fixes RNG stream; **no floats** appear in manifest.

## **10) Edge cases & failure policy**

Missing Registry/ParameterSet; both or neither of Ballots/Tally; duplicate kinds; wrong canonicalization tag; seed not 32-byte hex; non-64-hex sha256; negative length. **Error** and halt before run.

## **11) Test checklist (must pass)**

Valid minimal manifest passes; malformed cases hit the right validation errors.

After IO hash-verification, pipeline runs and later **RunRecord** echoes engine, formula\_id, IDs, and rng\_seed.