# **Pre-Coding Essentials (Component: schemas/manifest.schema.json, Version/FormulaID: VM-ENGINE v0) — 17/89**

## **1) Goal & Success**

Goal: JSON Schema for the **run manifest** that names the input artifacts and optional expectations (FormulaID/EngineVersion) for a deterministic, offline run.

Success: Validates **exactly one** ballots source (ballots *or* ballot\_tally), requires DivisionRegistry and ParameterSet, rejects URLs, and is strict (additionalProperties: false). Loader can resolve paths relative to the manifest file.

## **2) Scope**

In scope: Top-level metadata, relative file paths to inputs, optional expectations, optional file digests.

Out of scope: Algorithm behavior, hashing of Results/RunRecords (done by engine), cross-file integrity beyond presence (loader/pipeline validate those).

## **3) Inputs → Outputs**

Inputs: manifest.json.

Outputs: Pass/fail against schema; on pass, loader builds a LoadedContext (registry + options + tallies/ballots + params + expectations).

## **4) Entities/Fields (schema shape to encode)**

**Root object**

id **(required, string)** — arbitrary MAN:<name>:v<digits> (stable label for the manifest itself)

reg\_path **(required, string)** — path to division\_registry.json

params\_path **(required, string)** — path to parameter\_set.json

**Ballot source (exactly one required):**

ballots\_path **(string)** — path to raw ballots.json

ballot\_tally\_path **(string)** — path to aggregated ballot\_tally.json

**Optional inputs:**

adjacency\_path *(string)* — if adjacency is delivered separately

**Optional expectations (sanity locks, not normative):**

expect *(object)*:

formula\_id *(string)* — expected FormulaID of the rule set

engine\_version *(string)* — expected engine version

**Optional input digests (pre-flight integrity):**

digests *(object)* — map of { "<relative\_path>": { "sha256": "<hex64>" } }

notes *(optional, string)*

All paths are **relative to the manifest file** unless absolute. URLs are disallowed.

## **5) Variables (validators & enums used in schema)**

## **6) Functions**

(Schema only.)

## **7) Algorithm Outline (schema authoring steps)**

$schema = JSON Schema **2020-12**; set $id for stable tooling.

$defs: ManifestId, LocalPath, Sha256, Expect.

Root: type: object, required: ["id","reg\_path","params\_path"], additionalProperties: false.

**One-of**: encode oneOf requiring **exactly one** of ballots\_path or ballot\_tally\_path.

Paths: type: string, pattern: path.pattern.local. (No URLs.)

digests: type: object, additionalProperties schema { type: object, required: ["sha256"], properties: { sha256: { pattern: digest.hex64 }}, additionalProperties: false }.

expect: optional object with formula\_id and engine\_version as plain strings; no coupling enforced at schema-level (engine will compare at load time).

Strictness: every object sets additionalProperties: false.

## **8) State Flow**

Loader resolves manifest directory → joins relative paths → canonicalizes → schema-validate targets (registry, params, ballots/tally, adjacency if present) → builds LoadedContext for the pipeline.

## **9) Determinism & Numeric Rules**

Canonicalization rules (UTF-8, LF, sorted JSON keys; UTC timestamps) apply to all artifacts; manifest enforces **local files only** and allows optional digests for integrity.

No numeric computation here.

## **10) Edge Cases & Failure Policy**

Both ballots\_path **and** ballot\_tally\_path present → **schema fail**.

Neither present → **schema fail**.

Any \*\_path starting with http:// or https:// → **schema fail**.

digests present but hex length ≠ 64 or non-hex → **schema fail**.

expect.formula\_id mismatch at runtime → loader must **error** (fail fast before running).

Relative paths outside repo via .. are allowed at schema level; loader should resolve and may **reject** traversal if policy requires.

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

**Happy path (tally)**: has reg\_path, params\_path, ballot\_tally\_path; no ballots\_path → **pass**.

**Happy path (raw ballots)**: has reg\_path, params\_path, ballots\_path; no ballot\_tally\_path → **pass**.

Both ballot sources present → **fail**.

URL in any path → **fail**.

digests with bad hex → **fail**.