# **Addendum 1A — Formula ID & Canonical Serialization (Normative)**

**Status:** Normative. Required for all releases and for Tests VM-TST-019/020.  
 **Purpose:** Define exactly **what is hashed** to produce the **Formula ID** and **how** all canonical artifacts are serialized so byte-identical results are possible across platforms.

## **0) Definitions**

* **Formula ID (FID):** Cryptographic fingerprint of the **normative rule set** (NOT of any specific dataset or run).
* **Canonical Serialization:** Deterministic byte representation used for the FID **and** for hashing Results/RunRecords.
* **Normative Manifest (NM):** Machine-readable bundle enumerating rule primitives that affect outcomes.

## **1) What the Formula ID covers**

### **1.1 Variables that affect outcomes (IDs)**

Include the **existence**, **domain/semantics**, and **default value** for:

* **Ballot:** 001–007
* **Allocation & MMP:** 010–017
* **Gates & Families:** 020–029 + **021\_scope**
* **Aggregation:** 030–031
* **Frontier & Contiguity:** 040–048
* **Ties & RNG:** 050–052
* **Labels:** 060–062
* **Executive toggle:** 073

Exclude operational/presentation toggles that do not change outcomes (e.g., table sorting, report precision). Their defaults do **not** enter the FID.

### **1.2 Fixed algorithmic rules (constants)**

Include the following constants (they are part of the FID):

* **Approval gate denominator:** approvals\_for\_change / valid\_ballots.
* **IRV exhaustion policy:** reduce\_continuing\_denominator.
* **Rounding for comparisons:** round-half-to-even at the explicitly defined decision points.
* **Allowed allocation families:** winner\_take\_all, proportional\_favor\_big (D’Hondt), proportional\_favor\_small (Sainte-Laguë), largest\_remainder, mixed\_local\_correction (MMP).
* **MMP sequencing:** local seats → target shares → deficit calculation → top-ups (per **mlc\_correction\_level**) → overhang handling per **overhang\_policy/total\_seats\_model**.
* **Contiguity edge types:** {land, bridge, water} and their semantics.

### **1.3 What the FID does not cover**

* Data schemas (Doc 1), pipeline function names (Doc 5), report templates (Doc 7), performance profiles, UI text, translations.
* Any **run-time** parameter values chosen by users for a specific simulation (those appear in the RunRecord, not in the FID).

## **2) Building the Normative Manifest (NM)**

Construct a single JSON object with the following **canonical field order** (names exact):

1. "schema\_version" — string (e.g., "NM-1.0").
2. "variables" — array sorted by **VM-VAR ID** ascending; each item:  
   * "id" (e.g., "VM-VAR-022"),
   * "name" (stable snake\_case),
   * "domain" (closed set or numeric range),
   * "default",
   * "notes" (short semantics; no markdown).
3. "constants" — object with keys:  
   * "approval\_gate\_denominator", "irv\_exhaustion\_policy", "rounding\_rule",
   * "allocation\_families" (array, fixed order),
   * "mmp\_sequence" (array of step labels),
   * "contiguity\_edge\_types" (array).
4. "compat" — object with keys:  
   * "reserved\_ids" (arrays by range),
   * "fid\_policy\_version" (string).
5. "origin" — object (informative, **excluded from FID hash**, see §4.3):  
   * "docs\_commit\_refs" (map of doc→VCS ref), "generated\_at\_utc".

Only fields **1–4** are hashed for the FID. Field **5** is carried for traceability and is **explicitly excluded** from the FID computation.

## **3) Canonical Serialization Rules (apply to NM, Results, RunRecords)**

1. **Encoding:** UTF-8, **no BOM**, Unix line endings (\n).
2. **Whitespace:** JSON with a single space after colons and commas; no trailing spaces; no pretty alignment beyond that.
3. **Key ordering:**
   * Objects: keys sorted **lexicographically (UTF-8 code point)**.
   * Arrays:  
     + If representing **sets** (e.g., variable registry), sort by the specified key (ID ascending).
     + If representing **sequences** (e.g., MMP steps, ranked rounds), preserve declared order.
4. **Numbers:**
   * Integers: base-10, no leading + or zero padding.
   * Decimals: use the shortest representation that round-trips; scientific notation **disallowed**.
5. **Booleans/null:** JSON true/false/null (lowercase).
6. **Strings:** Normalize to **Unicode NFC**; escape only per JSON standard; no trailing \n.
7. **Dates/times:**
   * Dates: YYYY-MM-DD.
   * Timestamps: YYYY-MM-DDTHH:MM:SSZ (UTC only).
8. **Omissions:** Omit fields that are optional and unset; do not emit null in their place.
9. **Unit/Option ordering in artifacts:**
   * Units: sort by **Unit ID** (lexicographic).
   * Options: sort by **Option.order\_index** then **Option ID**.

## **4) Hashing & Identifiers**

### **4.1 Algorithm**

* **SHA-256** over the canonical byte stream.

### **4.2 Representations**

* **Formula ID (full):** 64 hex chars, lowercase.
* **Formula ID (short):** first **24** hex chars of the full (12 bytes), printed in report footers.
* **Result/RunRecord hash:** same algorithm and canonicalization, full 64-hex printed in RunRecord; report may show short form.

### **4.3 Exactly what is hashed**

* **FID hash input:** Canonical serialization of NM fields **schema\_version**, **variables**, **constants**, **compat** (in that object/key order and with the global rules in §3).
* **Excluded from FID:** "origin" block, any VCS refs, timestamps, file paths.
* **Result/RunRecord hash input:** Canonical serialization of the full Result/RunRecord objects, including:  
  + Registry and tally checksums/IDs,
  + ParameterSet values used for the run,
  + Engine Version, Formula ID (full), RNG seed(s), tie policy, determinism flags, and environment fingerprints as specified in Doc 3B.

## **5) Change Policy (when to bump FID vs Engine Version)**

**Bump Formula ID (FID)** when **any** of the following change:

* Add/remove a **VM-VAR** in the included ranges, change a **default**, change a **domain/semantics**.
* Modify any **constant** listed in §1.2 (denominators, rounding, exhaustion, allowed families, MMP sequence, contiguity semantics).
* Alter canonicalization rules in §3.

**Bump Engine Version only** when:

* Performance, packaging, UI/report wording, translations, or non-normative pipeline details change.
* Bug fixes that **do not** alter computed outcomes (for any allowed ParameterSet) and do not change §1.2 constants.

**Both must bump** if:

* A bug fix alters outcomes for any permitted ParameterSet (even if “correcting” to intent). Treat as a normative change → new FID and new Engine Version.

## **6) Compliance Hooks (tests)**

* **VM-TST-019:** Same OS repeated runs → **identical Result/RunRecord hashes** using §3 and §4.
* **VM-TST-020:** Cross-OS (Windows/macOS/Linux) → **identical hashes**; any discrepancy indicates a canonicalization violation or non-determinism.

## **7) Printing & Verification**

* Reports print: **Formula ID (short)**, **Engine Version**, and a notice if defaults differ from Annex A.
* RunRecord includes: **Formula ID (full)**, the NM digest section ("schema\_version", ranges covered), ParameterSet used, seed, and environment fingerprint.
* Verifiers recompute the FID from the embedded NM (fields §2.1–§2.4) and must obtain the same 64-hex value.

**End of Addendum 1A (Normative).**