# **Pre-Coding Essentials (Component: .pre-commit-config.yaml, Version/FormulaID: VM-ENGINE v0) — 8/89**

## **1) Goal & Success**

Goal: Fast, local gate that blocks non-canonical diffs and obvious policy drift before commit/push.

Success: Commits are LF-only, formatted, schema-valid; pushes run lint/tests and a tiny determinism smoke check.

## **2) Scope**

In scope: pre-commit (fast) and pre-push (heavier) checks for formatting, whitespace/EOL, schemas/fixtures, Rust fmt, optional quick tests.

Out of scope: full CI matrix; release packaging.

## **3) Inputs → Outputs**

Inputs: workspace sources, JSON schemas, fixtures, Cargo workspace, optional vm\_cli binary for canonical checks.

Outputs: Rejected commits/pushes on violations; clean committed diffs.

## **4) Entities/Tables**

## **5) Variables**

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

(YAML config; hooks call tools. No Rust functions.)

## **7) Algorithm Outline (hook plan)**

**Pre-commit (fast, <3–5s typical):**

**LF & whitespace**: run end-of-file-fixer, trailing-whitespace (exclude \*.md), and mixed-line-ending (ban CRLF).

**JSON sanity**: check-json on staged \*.json; check-merge-conflict.

**Schemas**: basic lint (well-formed JSON) for schemas/\*\*/\*.json.

**Rust format**: cargo fmt --all -- --check (fail on diff).

**Canonical JSON check** (when available): vm\_cli canonicalize --check <staged-json> to assert sorted keys + LF.

**Pre-push (heavier):** 6) **Clippy**: cargo clippy --all-targets -- -D warnings.  
 7) **Unit tests**: cargo test --locked --workspace.  
 8) **Determinism smoke**: run one tiny Annex B fixture twice with --rng-seed $determinism.seed; compare RES:/RUN: IDs.

*All hooks should be implemented as* ***local*** *hooks to avoid network fetch; keep them shell-agnostic where possible.*

## **8) State Flow**

Developer stages changes → pre-commit runs fast checks → commit allowed or rejected.

On push → heavier checks run; push blocked on failures.

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

Enforce LF-only, sorted JSON keys (via canonicalize check), and fixed RNG seed in smoke test to catch nondeterminism.

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

Windows: prefer invoking via bash (Git for Windows) or use language: system local hooks without bashisms.

If vm\_cli isn’t built yet, skip canonicalize step (guard with command\_exists); CI will still enforce in pipeline.

Keep pre-commit fast; move anything >5s to pre-push/CI.

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

Editing JSON/RS triggers formatter/JSON checks; commit blocked on violations.

Introducing CRLF in a file → commit blocked.

Pushing with a clippy error or failing test → push blocked.

Determinism smoke: two runs produce identical IDs; otherwise push blocked.

### **Suggested .pre-commit-config.yaml skeleton (for later implementation)**

yaml

CopyEdit

repos:

# Basic hygiene (use local mirroring where possible)

- repo: local

hooks:

- id: eof-fixer

name: end-of-file-fixer

entry: bash -c 'python - "$@" << "PY"\nimport sys, pathlib\n# minimal EOF fixer for staged files\nPY'

language: system

files: '.\*' # keep placeholder; replace with real fixer or pre-commit-hooks when allowed

- id: trailing-whitespace

name: trailing-whitespace

entry: bash -c 'grep -nI "[[:blank:]]$" $@ && exit 1 || exit 0'

language: system

files: '^(?!.\*\\.md$).\*'

- id: mixed-line-ending

name: mixed-line-ending (ban CRLF)

entry: bash -c 'grep -IUr --files-with-matches "\r" $@ && exit 1 || exit 0'

language: system

files: '.\*'

- id: check-json

name: check-json

entry: bash -c 'for f in "$@"; do jq -e . "$f" >/dev/null || exit 1; done'

language: system

files: '\\.json$'

- id: cargo-fmt

name: cargo fmt --check

entry: bash -c 'cargo fmt --all -- --check'

language: system

pass\_filenames: false

stages: [commit]

- repo: local

hooks:

- id: cargo-clippy

name: cargo clippy -D warnings

entry: bash -c 'cargo clippy --all-targets -- -D warnings'

language: system

pass\_filenames: false

stages: [push]

- id: cargo-test

name: cargo test --locked

entry: bash -c 'cargo test --locked --workspace'

language: system

pass\_filenames: false

stages: [push]

- id: determinism-smoke

name: determinism smoke (double run one fixture)

entry: bash -c 'set -e; vm\_cli run --manifest fixtures/annex\_b/part\_0/manifest.json --rng-seed "$VM\_SEED" --out artifacts/run1 && vm\_cli run --manifest fixtures/annex\_b/part\_0/manifest.json --rng-seed "$VM\_SEED" --out artifacts/run2 && diff <(jq -r .id artifacts/run1/result.json) <(jq -r .id artifacts/run2/result.json)'

language: system

pass\_filenames: false

stages: [push]

Note: Replace placeholder hygiene hooks with official pre-commit-hooks repo when you’re ready to allow downloads, or keep using local commands for fully offline operation.