**THE BIBLE FOR M0 TO M5**

Absolutely—let’s lock down how **M0→M5** runs today so anyone (including a new chat, auditor, or human coder) can reproduce it safely.

Below I’m giving you:

1. a **copy‑pasteable runbook** (docs/OPERATIONS\_RUNBOOK\_M0\_M5.md),
2. two **data‑contract docs** for M3 and M4 (docs/m3\_contract.md, docs/m4\_contract.md), and
3. a **tiny CI gate script** (tools/ci\_gate.ps1) that re‑runs the green path, validates contracts, and runs smoke tests.

Everything is written to match what *you already ran successfully* (your smoke reports and the contract validator output).

**1) docs/OPERATIONS\_RUNBOOK\_M0\_M5.md**

# TerraNova – Operations Runbook (M0 → M5)

\*\*Status:\*\* Stable baseline (Windows, PowerShell, Python 3.12, venv)

\*\*Modules covered:\*\* M0 Setup → M1 Ops → M2 Working Capital & P/L → M3 Financing → M4 Tax → M5 Cash Flow (CFO)

---

## A. Directory layout (top-level)

C:\TerraNova  
├─ .venv\ # venv used to run everything  
├─ src\terra\_nova\modules\ # M0..M5 live code used by orchestrator + M5 runner  
│ ├─ m0\_setup  
│ ├─ m1\_operational\_engines  
│ ├─ m2\_working\_capital\_pl  
│ ├─ m3\_financing\ # restored module wrapper for imports  
│ ├─ m4\_tax  
│ └─ m5\_cash\_flow\runner.py # M5 "accept either" logic (inspect/full run)  
├─ baselines  
│ ├─ TerraNova\_main\_up\_to\_M2\ # stable, used by M3 imports  
│ ├─ TerraNova\_M3\_FinancingEngine\_v3.1.0  
│ └─ TerraNova\_M4\_TaxEngine\_v4.0.3  
├─ docs\ # contracts + technical notes  
├─ tools\ # validators, helpers  
├─ tests\ # smoke tests  
├─ InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx  
└─ outputs\ # all artifacts (.parquet, .md, debug)

\*\*Key artifacts (current naming):\*\*

- M2:

- `m2\_pl\_schedule.parquet`

- `m2\_working\_capital\_schedule.parquet`

- M3:

- `m3\_financing\_schedule.parquet`

- `m3\_summary.parquet`

- M4:

- `m4\_tax\_schedule.parquet`

- `m4\_tax\_summary.parquet`

- M5:

- `m5\_cash\_flow\_statement\_final.parquet` (CFO)

- `m5\_smoke\_report.md`

- `m5\_debug\_dump.json` (optional)

- General:

- `m0\_smoke\_report.md` … `m5\_smoke\_report.md`

- `contracts\_validate\_debug.json` (tools\validate\_contracts.py)

---

## B. Environment prerequisites (one‑time)

From \*\*PowerShell\*\* in `C:\TerraNova`:

```powershell

# 1) Activate venv

.\.venv\Scripts\Activate.ps1

# 2) Make src importable permanently inside the venv

.\.venv\Scripts\python.exe -c "import site,pathlib;sp=[p for p in site.getsitepackages() if p.endswith('site-packages')][0];pathlib.Path(sp,'terranova\_src.pth').write\_text(r'C:\TerraNova\src');print('PYTHONPATH pinned at',pathlib.Path(sp,'terranova\_src.pth'))"

This creates .venv\Lib\site-packages\terranova\_src.pth so imports like terra\_nova.modules.\* always resolve without setting PYTHONPATH by hand.

**C. Fresh run – the known‑good path**

Use these exact one‑liners. They’re idempotent and match what we validated.

**M3:**

$env:PYTHONPATH="C:\TerraNova\src;C:\TerraNova\baselines\TerraNova\_main\_up\_to\_M2\src"; .\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M3\_FinancingEngine\_v3.1.0\run.py" run\_m3 --input "C:\TerraNova\InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx" --out "C:\TerraNova\outputs" --currency NAD

**M4:**

.\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M4\_TaxEngine\_v4.0.3\run.py" run\_m4 --input "C:\TerraNova\InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx" --out "C:\TerraNova\outputs" --currency NAD

**M5 – inspect‑only (just checks discovery & mapping):**

.\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'.\outputs','NAD',inspect\_only=True)"

**M5 – full run:**

.\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'.\outputs','NAD',inspect\_only=False)"

**Contract validator (M2 contract):**

.\.venv\Scripts\python.exe .\tools\validate\_contracts.py .\outputs

**Smoke tests:**

.\.venv\Scripts\python.exe -m unittest -v tests\smoke\test\_pipeline\_smoke.py

**D. What the modules output (columns we rely on)**

These are taken from the smoke reports produced on the stable run.

**M2 P&L (file: m2\_pl\_schedule.parquet)**

Month\_Index,

Total\_Revenue\_NAD\_000, Variable\_OPEX\_NAD\_000, Fixed\_OPEX\_NAD\_000, Total\_OPEX\_NAD\_000,

Depreciation\_NAD\_000, EBITDA\_NAD\_000, EBIT\_NAD\_000,

Interest\_Expense\_NAD\_000, PBT\_NAD\_000, Tax\_Expense\_NAD\_000, NPAT\_NAD\_000

**M2 Working Capital (file: m2\_working\_capital\_schedule.parquet)**

Month\_Index,

Change\_in\_Receivables\_NAD\_000, Change\_in\_Inventory\_NAD\_000, Change\_in\_Payables\_NAD\_000,

Cash\_Flow\_from\_NWC\_Change\_NAD\_000

**M3 Financing schedule (m3\_financing\_schedule.parquet)**

Month\_Index,

Debt\_Begin\_NAD\_000, New\_Borrowing\_NAD\_000,

Principal\_Repayment\_NAD\_000, Interest\_NAD\_000,

Debt\_End\_NAD\_000, Equity\_Infusion\_NAD\_000

**M4 Tax schedule (m4\_tax\_schedule.parquet)**

Month\_Index, Taxable\_Income\_NAD\_000, Tax\_Rate,

Tax\_Expense\_NAD\_000, Tax\_Paid\_NAD\_000, Tax\_Payable\_End\_NAD\_000

**M5 Cash Flow (m5\_cash\_flow\_statement\_final.parquet)**

Month\_Index,

Net\_Profit\_After\_Tax\_NAD\_000, Depreciation\_NAD\_000, WC\_Cash\_Flow\_NAD\_000,

Interest\_Paid\_NAD\_000, Tax\_Paid\_NAD\_000, Cash\_Flow\_from\_Operations\_NAD\_000

**M5 formula (verified in smoke):**

CFO = NPAT + Depreciation + WC\_CF − Tax\_Paid − Interest\_Paid

**M2→M5 role mapping (normalized inside M5):**

* DA ← Depreciation\_NAD\_000
* NPAT ← NPAT\_NAD\_000
* NWC\_CF ← Cash\_Flow\_from\_NWC\_Change\_NAD\_000
* MONTH\_INDEX ← Month\_Index

**E. Backups**

We use robocopy to snapshot code + outputs:

$ts = Get-Date -Format "yyyyMMdd\_HHmm"

robocopy "C:\TerraNova" "D:\TN\_Backups\TerraNova\_$ts" /MIR /XD ".venv" "workspace" "drops" ".git" /XF \*.pyc

**F. Troubleshooting (fast)**

* **ModuleNotFoundError: terra\_nova**  
  Re‑pin src:  
  .\.venv\Scripts\python.exe -c "import site,pathlib;sp=[p for p in site.getsitepackages() if p.endswith('site-packages')][0];pathlib.Path(sp,'terranova\_src.pth').write\_text(r'C:\TerraNova\src')"
* **M5 inspect says cannot find M2 files**  
  Ensure m2\_pl\_schedule.parquet and m2\_working\_capital\_schedule.parquet exist in .\outputs.  
  If they’re under baselines\TerraNova\_main\_up\_to\_M2\outputs, copy them:  
  Get-ChildItem "C:\TerraNova\baselines\TerraNova\_main\_up\_to\_M2\outputs" -Filter "m2\_\*schedule\*.parquet" | Copy-Item -Destination "C:\TerraNova\outputs" -Force
* **Contract validator fails for M2**  
  Run M3 & M4 again in case upstream refreshed a column; re‑run tools\validate\_contracts.py .\outputs.
* **Change control**  
  Do **not** rename artifacts without updating the corresponding contract doc and (if needed) M5 mapping.

**G. Release practice**

1. Update docs/contracts.
2. Run M3 → M4 → M5 → contract validator → smoke tests.
3. When green, commit + tag (e.g., m0m5-stable-v1).
4. Keep zips out; package code in the repo.

---

## 2) `docs/m3\_contract.md` (freeze the interface you actually produce)

```markdown

# M3 Contract (Financing Engine) – v1

\*\*Artifact(s):\*\*

- `outputs/m3\_financing\_schedule.parquet`

- `outputs/m3\_summary.parquet`

\*\*Required columns – schedule (downstream dependencies):\*\*

Month\_Index,  
Debt\_Begin\_NAD\_000, New\_Borrowing\_NAD\_000,  
Principal\_Repayment\_NAD\_000, Interest\_NAD\_000,  
Debt\_End\_NAD\_000, Equity\_Infusion\_NAD\_000

\*\*Semantics:\*\*

- `Month\_Index`: 1..N (continuous)

- Interest is \*\*cash\*\* interest paid that period (links to M5 `Interest\_Paid\_NAD\_000`)

- `Principal\_Repayment\_NAD\_000`: cash principal paid

- `Equity\_Infusion\_NAD\_000`: cash equity injections

\*\*Guarantees:\*\*

- `Debt\_End\_NAD\_000 = Debt\_Begin + New\_Borrowing − Principal\_Repayment` (per month)

- All currency fields in NAD thousands.

\*\*Versioning:\*\*

- Column names are \*\*frozen\*\*; any rename requires updating this contract and all downstream readers.

**3) docs/m4\_contract.md (freeze the interface you actually produce)**

# M4 Contract (Tax Engine) – v1

\*\*Artifact(s):\*\*

- `outputs/m4\_tax\_schedule.parquet`

- `outputs/m4\_tax\_summary.parquet`

\*\*Required columns – schedule (downstream dependencies):\*\*

Month\_Index,  
Taxable\_Income\_NAD\_000, Tax\_Rate,  
Tax\_Expense\_NAD\_000, Tax\_Paid\_NAD\_000, Tax\_Payable\_End\_NAD\_000

\*\*Semantics:\*\*

- `Tax\_Expense\_NAD\_000`: accrual tax for the month

- `Tax\_Paid\_NAD\_000`: cash tax paid in the month (feeds M5)

- `Tax\_Payable\_End\_NAD\_000`: closing payable balance

\*\*Guarantees:\*\*

- Currency fields in NAD thousands.

- Month index is continuous and matches M2/M3.

\*\*Versioning:\*\*

- Column names are \*\*frozen\*\*; any rename requires updating this contract and downstream logic.

**4) tools/ci\_gate.ps1 (poor‑man’s CI – one command)**

Save as tools/ci\_gate.ps1:

param(

[string]$OutDir = "C:\TerraNova\outputs",

[string]$InputXlsx = "C:\TerraNova\InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx"

)

$ErrorActionPreference = "Stop"

Write-Host "== CI Gate: M3 -> M4 -> M5 -> Contracts -> Smoke"

# Ensure venv

if (-not (Test-Path ".\.venv\Scripts\python.exe")) { throw "venv missing" }

# Pin src to site-packages (idempotent)

.\.venv\Scripts\python.exe -c "import site,pathlib;sp=[p for p in site.getsitepackages() if p.endswith('site-packages')][0];pathlib.Path(sp,'terranova\_src.pth').write\_text(r'C:\TerraNova\src')"

# M3 (needs M0..M2 src on path for imports)

$env:PYTHONPATH="C:\TerraNova\src;C:\TerraNova\baselines\TerraNova\_main\_up\_to\_M2\src"

.\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M3\_FinancingEngine\_v3.1.0\run.py" run\_m3 --input $InputXlsx --out $OutDir --currency NAD

# M4

$env:PYTHONPATH="C:\TerraNova\src"

.\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M4\_TaxEngine\_v4.0.3\run.py" run\_m4 --input $InputXlsx --out $OutDir --currency NAD

# M5 inspect + full

.\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'$OutDir','NAD',inspect\_only=True)"

.\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'$OutDir','NAD',inspect\_only=False)"

# Contracts + smoke

.\.venv\Scripts\python.exe .\tools\validate\_contracts.py $OutDir

.\.venv\Scripts\python.exe -m unittest -v tests\smoke\test\_pipeline\_smoke.py

Write-Host "== CI Gate: PASS"

Usage (one line):

powershell -ExecutionPolicy Bypass -File .\tools\ci\_gate.ps1

**Do we need to “wire” a new top‑level run.py now?**

**Short answer: no.**  
Your current run.py is working and you’ve locked the interfaces with contracts + smoke tests. For M6–M9 development this modular setup is actually **safer**:

* We call M3/M4 via their baseline run.py scripts (proven good).
* M5 is a thin, importable runner.py under src that **normalizes** inputs and computes CFO using the accepted name set.
* We gate every change with **ci\_gate.ps1** (M3→M4→M5→contracts→smoke).

When we finish M6–M9 (Balance Sheet, etc.), we can **optionally** refactor into a single cohesive run.py that delegates to terra\_nova.modules.\*.runner.run\_mX functions. That refactor is purely convenience—*not* a functional requirement for progress on M6.

**What this buys us (for the hand‑over)**

* A **repeatable sequence** of one‑liners & a CI script.
* **Frozen contracts** for M2 (validated), and M3/M4 (documented now to be frozen).
* **Detectable regressions**: smoke tests + contract validator fail fast.
* **No zip packaging**—we ship code, not archives.
* **Clear troubleshooting** steps (path pinning, artifact discovery, contract mapping).

If you want, I can also provide a tiny patch to extend tools\validate\_contracts.py to validate M3/M4 columns against the two new contracts (same style as M2). But given your current green state, you can proceed to M6 and add those checks right before you cut the first M6 tag.

**Quick command crib (copy/paste)**

* **M3:**  
  $env:PYTHONPATH="C:\TerraNova\src;C:\TerraNova\baselines\TerraNova\_main\_up\_to\_M2\src"; .\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M3\_FinancingEngine\_v3.1.0\run.py" run\_m3 --input "C:\TerraNova\InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx" --out "C:\TerraNova\outputs" --currency NAD
* **M4:**  
  .\.venv\Scripts\python.exe "C:\TerraNova\baselines\TerraNova\_M4\_TaxEngine\_v4.0.3\run.py" run\_m4 --input "C:\TerraNova\InputPack\TerraNova\_Input\_Pack\_v10\_0.xlsx" --out "C:\TerraNova\outputs" --currency NAD
* **M5 (inspect):**  
  .\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'.\outputs','NAD',inspect\_only=True)"
* **M5 (full):**  
  .\.venv\Scripts\python.exe -c "from terra\_nova.modules.m5\_cash\_flow.runner import run\_m5; run\_m5(r'.\outputs','NAD',inspect\_only=False)"
* **Contracts:**  
  .\.venv\Scripts\python.exe .\tools\validate\_contracts.py .\outputs
* **Smoke:**  
  .\.venv\Scripts\python.exe -m unittest -v tests\smoke\test\_pipeline\_smoke.py
* **CI gate:**  
  powershell -ExecutionPolicy Bypass -File .\tools\ci\_gate.ps1

**Next step toward M6 (when you’re ready)**

* Keep this runbook + contracts in docs/.
* Extend tools\validate\_contracts.py to read docs/m3\_contract.md & docs/m4\_contract.md (or bake the column lists inline) and assert those columns before we start Balance Sheet.
* Define the **M6 input contract** from the columns listed above (M2/M3/M4/M5) and the **M6 output contract** (balance sheet columns, and “Assets = Liabilities + Equity” check).
* Follow the same discipline: implement → validate → smoke → tag.

If you want me to supply the small validate\_contracts.py patch for M3/M4, say the word and I’ll provide the exact diff you can paste in.