Forensic CPM Engine — Reproducibility Certificate

One-page expert-disclosure attachment. The engine that produced your forensic report is independently reproducible by any opposing expert.

VERSION 2.9.16 · SHA-256-LOCKED · SIGSTORE-SIGNED

ENGINE IDENTITY

JAVASCRIPT ENGINE SHA-256
`4815ba956d733ea4c008da4337bf2fa36c7d9ef3630a027475e28a4d2bea5e0d`

PYTHON REFERENCE SHA-256
`1537dfd490062bfe1b726f86cabf4731d8a2218b6013732c8a53256a2bcb7e7c`

GITHUB COMMIT (TAG V2.9.16)
`c0af428c9b6465f073f5d00d2d0f4a80507cc69f`

TOPOLOGY-HASH FORMAT
`sha256-canonical-v2 (prefix v2:)`

FOUR PUBLIC DEPLOYMENT SURFACES — BIT-IDENTICAL

SURFACE	INSTALL PATH	VERIFICATION
npm package	`npm install cpp-cpm-engine@2.9.16`	`npm view cpp-cpm-engine@2.9.16 dist.shasum`
GitHub release	`github.com/danafitkowski/cpp-cpm-engine/releases/tag/v2.9.16`	`gh attestation verify cpp-cpm-engine-2.9.16.tgz`
Railway-hosted MCP	`curl mcp.criticalpathpartners.ca/cpm-engine.js`	`sha256sum — compare to cell above`
Local skill bundle	`~/.claude/skills/cpp-forensic-mcp/static/cpm-engine.js`	`sha256sum — compare to cell above`
MCP registry index	`registry.modelcontextprotocol.io/v0/servers?search=cpp-cpm`	`version === 2.9.16, isLatest: true`

Reproduction in under a minute, zero npm dependencies, any opposing expert

```
git clone https://github.com/danafitkowski/cpp-cpm-engine
cd cpp-cpm-engine && git checkout v2.9.16
npm test                # 878 unit tests
node cpm-engine.crossval.js   # 43 fixtures × 444 checks JS⇔Python parity
```

VERIFIABLE EVIDENCE

UNIT TESTS PASSING
878 / 878 assertions

CROSS-VALIDATION
43 fixtures · **444** / 444 checks

CI MATRIX
3 OS × 3 Node = **9 cells** per push

ENGINE LINES OF CODE
JS **6,950** · Python **1,304**

HOLIDAY CALENDARS
66 jurisdictions (CA+13, US+50, +UK)

SIGSTORE SIGNATURE
Rekor public transparency log

AACE-CANONICAL METHOD IDENTIFIERS

- **AACE 29R-03** — Forensic Schedule Analysis. MIP 3.3 Observational Windows; MIP 3.6 Prospective Single-Base TIA; MIP 3.7 Prospective Multi-Base TIA; MIP 3.8 Collapsed As-Built (Single Sim).
- **AACE 49R-06** — Critical Path identification methods (LPM · TFM · MFP).
- **AACE 52R-06** — Time Impact Analysis (fragnet insertion).
- **AACE 24R-03** — Constraint criticality.
- **AACE 67R-11** — Forensic schedule analysis competency.
- **DCMA 14-Point Assessment** — full coverage in CI.
- **SCL Delay & Disruption Protocol, 2nd Edition (2017)** — methodology alignment.

DAUBERT FRAMEWORK COMPLIANCE

- **FRE 702** as amended December 1, 2023 — methodology reliability (Rule 702(c) and 702(d)) demonstrated to preponderance of evidence. All four *Daubert* prongs addressed in `buildDaubertDisclosure()`.
- ***Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 509 U.S. 579 (1993)** — foundational reliability framework: testability, peer review, error rate, general acceptance.
- **FRCP 26(a)(2)(B)** — expert disclosure of methods and data. Engine version + topology hash + manifest fields satisfy disclosure.
- **Proposed FRE 707** — court-appointed-expert methodology inspectability. Forward-compatible.

Dana Fitkowski
Forensic Construction Schedule Claims Expert · Critical Path Partners
25 years P6 · 20 years nuclear & heavy energy

criticalpathpartners.ca
dana@criticalpathpartners.ca · +1-519-532-6300
Issued 2026-05-17 · v2.9.16 · MIT