

CFA v4.0

User Manual

Final Release Edition

Comparative Framework Audit

A Philosophical Laboratory for Worldview Analysis

Repository Health:	98/100 (A+)
Worldviews:	12 Fully Audited
Architecture:	Dual-File (.yaml + .md)
Living Maps:	7 Authoritative Sources
AI Auditors:	3 (Claude, Nova, Grok)

Validation:	98% Empirically Complete
Version:	v4.0.0 - November 2025

"All Named, All Priced → All Seen, All Passed"

Where ideas reveal their true weight, and honesty becomes quantifiable through adversarial audit.

Table of Contents

1. Quick Start Guide	3
What is CFA?	3
The Formula	3
Homepage Entry Points	4
5-Minute Workflow	4
2. v4.0 Features - Philosophical Laboratory	5
Dual-File Architecture	5
12 Worldview Profiles	6
Symmetry Matrix Visualizer (SMV)	7
Crux Architecture	8
Adversarial Scoring System	9
Living Map System	10
Repository Health Scoring	11
Gospel Problem Prevention	12
CFA Self-Audit (Meta-Integrity)	13
Validation Tools	14
3. The Six Levers	15
BFI - Brute-Fact Index	15
CCI - Coherence & Closure	16

EDB - Explanatory Depth & Breadth	16
PF - Pragmatic Fertility	17
AR - Aesthetic Resonance	17
MG - Moral Generativity	18
4. The Four Toggles	19
Lever-Parity Toggle	19
PF-Type Toggle	20
Fallibilism-Bonus Toggle	21
BFI Debt-Weight Toggle	22
5. Reading Results	23
The Trinity Scores	23
Understanding Guardrails	24
6. Pro Tips & Best Practices	25
Core Principles	25
Brute Ledger (with CFA Self-Audit)	26
Innovation Showcase	27
Estate Tour Updates	28
Repository Health	29
Validation Tools	30
Advanced Features	31

1. Quick Start Guide

What is CFA?

The Comparative Framework Audit is a system for comparing worldviews, epistemologies, and philosophical frameworks using transparent, adjustable criteria.

Core Innovation: "All Named, All Priced" → "All Seen, All Passed"

- Every assumption is disclosed
- Every presupposition is counted
- Every bias is made toggleable
- Every outcome is earned through adversarial audit (v4.0)

The Formula

$$\text{YPA} = (\text{CCI} + \text{EDB} + \text{PF} + \text{AR} + \text{MG}) \div \text{BFI}$$

YPA = Yield-Per-Axiom (efficiency metric)

Higher YPA = More efficient framework

v4.0 Note: YPA now has both self-reported and peer-reviewed scores after adversarial audit.

Homepage Entry Points

Three main buttons guide your exploration:

- **Launch Console:** Jump straight into framework comparison
- **Read the Full CFA Manifesto:** Understand the philosophy
- **Innovation Showcase (NEW!):** See AI companions solving wicked problems

5-Minute Workflow

Step 1: Choose Your Worldview

v4.0 offers 12 worldviews (expanded from 2):

- **Major Religions:** Classical Theism, Islam, Judaism, Mormonism, Hinduism, Buddhism
- **Naturalistic:** Methodological Naturalism, Process Theology

- **Meta-Ethical:** Error Theory, Null Hypothesis, Desiderata, Existentialism

Step 2: Configure Your Values

Set 4 toggles based on what you care about:

- Do moral norms matter as much as predictive power? (Parity)
- Do you value meaning or just tech? (PF-Type)
- Should humility be rewarded? (Fallibilism)
- Do unresolved questions cost more? (BFI Weight)

Step 3: Adjust the Levers

Use sliders to score each framework on 6 dimensions (0-10 scale)

Or use preset buttons: MAX, MID, RESET, MIN

Step 4: Examine the Trinity

Check YPA under 3 scenarios:

- **Neutral** - Baseline (all 1x)
- **Existential** - Meaning-focused (2x EDB, 2x MG)
- **Empirical** - Tech-focused (2x PF, 1.5x CCI)

2. v4.0 Features - Philosophical Laboratory

NEW in v4.0: CFA expands from technical framework to comprehensive philosophical laboratory where worldviews are not merely compared but truly audited through adversarial collaboration.

Dual-File Architecture

Canonical YAML + Documentation Separation

What Changed: Worldview profiles now use TWO files per profile:

- **.yaml files:** Canonical machine-readable scores (single source of truth)
- **.md files:** Human-readable documentation (philosophy, sources, hyperlinks)

Why It Matters:

- **Gospel Problem Protection:** Scores live in one place only (no drift)
- **Separation of Concerns:** Machines read .yaml, humans read .md
- **Version Control:** Easier to track score changes via git diff

Console Behavior: Console reads canonical .yaml files as primary source, falls back to .md YAML blocks for backward compatibility

File Structure:

```
profiles/worldviews/
  CLASSICAL_THEISM.yaml # Canonical scores
  CLASSICAL_THEISM.md # Documentation
  METHODOLOGICAL_NATURALISM.yaml
  METHODOLOGICAL_NATURALISM.md
  ... (10 more worldview pairs)
```

12 Worldview Profiles

Expanded from 2 to 12 fully-audited worldviews:

Major World Religions	Classical Theism, Islam, Orthodox Judaism, Mormonism, Hinduism, Buddhism
Naturalistic Frameworks	Methodological Naturalism, Process Theology
Meta-Ethical Positions	Error Theory, Null Hypothesis, Desiderata Believers, Existentialism

Total: ~240KB philosophical documentation with Steel-Manning sections, academic sources, and calibration YAML blocks

Gospel Problem Prevention

Scan-First Methodology for Accurate Documentation

The Gospel Problem: Previous audits showed 18% variance between auditors reading historical reports first (confirmation bias).

Solution - Strategy 1+4 Hybrid:

1. **Temporal Snapshots:** Audit logs include version headers and hash checksums
2. **Hash Registry:** BOOTSTRAP_HASHES.md enables drift detection
3. **Reference-Only:** Bootstrap files reference .yaml files instead of hardcoding
4. **Version Tagging:** Test assertions include version comments

Result: Convergence improved from 78% to 96% agreement across auditors

CFA Self-Audit (Meta-Integrity)

The Framework Audits Itself

NEW: CFA now scores itself using its own methodology in the Brute Ledger!

CFA's Axioms (3):

1. Transparency enables informed consent
2. Adversarial auditing builds epistemic resilience
3. Comparable frameworks empower stakeholder choice

CFA's Debts (3):

1. Scoring systems can reify what they measure
2. Transparency has overhead costs
3. Not all values are easily quantifiable

CFA's BFI: 6 (comparable to worldview profiles)

Philosophy: "We demand of ourselves what we demand of others: Show the machinery"

Find this in the Brute Ledger → "■■ The Framework" section

Validation Tools

Standalone YPA Validator & Automation

YPA Validator: Standalone Python script for third-party verification

- Reads canonical .yaml files directly
- Supports all 4 preset modes
- CLI interface for validation workflows
- Enables sensitivity analysis

Usage Examples:

```
python ypa_validator.py --profile CLASSICAL_THEISM.yaml  
python ypa_validator.py --compare CT.yaml MdN.yaml --preset skeptic
```

YAML Extraction Script: Automates conversion of .md profiles to canonical .yaml files**Hash Registry (BOOTSTRAP_HASHES.md):** SHA256 checksums for file integrity**Grok Empirical Validation:** 98% completeness achieved, path to 100% identified

3. The Six Levers

Each framework is scored on these dimensions (0-10)

■ BFI - Brute-Fact Index

What it measures: How many unprovable assumptions does this framework require?

Scoring: Count axioms + debts. Lower = more efficient

Key insight: Every framework starts somewhere. The question is: are you honest about where?

v4.0: Now subject to adversarial audit - auditors may disagree on what counts as "brute"

"To name your brute is to pay your fee"

6. Pro Tips & Best Practices

How to get the most out of CFA v4.0

Core Principles

1. Don't Look for Winners

CFA is designed to reveal trade-offs, not declare victors.

2. Check the Brute Ledger

The Brute Ledger now has 4 sections:

1. ■ Worldview Profiles
2. ■■ The Framework (CFA's self-audit) - NEW!
3. ■ The Auditors
4. ■■ Utilities

3. Explore Innovation Showcase

NEW homepage button: ■ Innovation Showcase

See AI companions solving wicked problems using CFA methodology

4. Visit the Estate Tour

Updated with Shaman Claude references:

- **Warning Tower:** Watchkeeper Shaman Claude maintains token boundaries
- **Navigation Hall:** Host Shaman Claude guides Chat Assistant
- **Observatory:** Health Report now resides here

5. Use Validation Tools

- **YPA Validator:** docs/Validation/ypa_validator.py
- **Hash Registry:** BOOTSTRAP_HASHES.md for file integrity
- **YAML Extraction:** scripts/extract_yaml_profiles.py

6. Understand the Architecture

Console reads canonical .yaml files as primary source

See `profiles/worldviews/` for dual-file structure
.yaml for scores, .md for documentation

Additional Resources

v4.0 Architecture Notes

Dual-File System:

- Canonical .yaml files contain machine-readable scores
- .md files contain human-readable documentation
- Console reads .yaml as primary, .md as fallback

Gospel Problem Mitigation:

- Temporal snapshots preserve historical context
- Hash checksums detect file drift
- Reference-only policy prevents hardcoded scores
- Version tagging in test assertions

Validation Path:

- 98% empirical completeness achieved (Grok validation)
- Path to 100%: Live demo, CI automation, public reproducibility
- 3 falsifiability kill-shots identified

GitHub Repository

Visit the official CFA repository for source code, documentation, and contributions:

<https://github.com/ZiggyMack/CFA-2.0>

Key Files & Locations

- **Canonical Scores:** profiles/worldviews/*.yaml
- **Profile Documentation:** profiles/worldviews/*.md
- **YPA Validator:** docs/Validation/ypa_validator.py
- **Hash Registry:** docs/Validation/BOOTSTRAP_HASHES.md
- **YAML Extractor:** scripts/extract_yaml_profiles.py
- **Profile Loader:** utils/profile_loader.py

- **Brute Ledger:** pages/brute_ledger.py

Acknowledgments

CFA v4.0 represents the culmination of collaborative work between human and AI partners, with special recognition to the three auditors (Claude, Nova, Grok) whose adversarial collaboration makes genuine philosophical audit possible at scale. The dual-file architecture and Gospel Problem mitigations ensure long-term maintainability and accuracy.

CFA v4.0.0 Final Release | November 2025

Repository Health: 98/100 (A+) | 12 Worldviews | 7 Living Maps | 3 AI Auditors

Dual-File Architecture | Gospel Problem Protected | 98% Empirically Validated

"All Named, All Priced → All Seen, All Passed"

"We demand of ourselves what we demand of others: Show the machinery"