HCP Portfolio Tracker PRD v3.4.3

Version: 3.4.3
File: hcp_tracker_prd_v3_4_3.md
Last Updated: 2025-08-31 23:45:00 UTC
Status: CURRENT - Production Release Documentation

---

1. Overview

1.1 Project Mission

Create a sophisticated yet user-friendly portfolio allocation tracker that implements HCP (Historical Context-based Portfolio) methodology for systematic investment decision-making across macroeconomic scenarios.

1.2 Current Status (v6.4.3)

- **Steps 1-3:** Fully functional and tested
- **Philosophy:** Complete investment framework acknowledgment
- **Data Management:** 5 sample scenarios + full editing capability
- **Theme Analysis:** Real IPS v3.10 calculations with proper momentum
- **File Size:** 112KB single-file deployment
- **Architecture:** Hybrid modular development → integrated deployment

---

2. Architecture Framework

2.1 Development Philosophy

The HCP Tracker uses a **hybrid development architecture** balancing clean development practices with simple deployment requirements.

2.2 Modular Development Structure

```
/development
├── file_handler_v1_4.js       # Sample data generation, file I/O
├── theme_calculator_v2_9.js   # IPS v3.10 probability calculations
├── data_editor_v1_0.js        # Modal editing system with overrides
├── indicators_v1_0.js         # 13 indicators with tier weighting
├── optimizer_v1_0.js      # [Next: Mean-variance optimization]
└── integration_tests/     # Module compatibility testing
```

2.3 Embedded Modules Architecture

```javascript
// Embedded modules available globally
const FileHandler = { version: '1.4', /* methods */ };
const ThemeCalculator = { version: '2.9', /* methods */ };
const DataEditor = { version: '1.0', /* methods */ };
const Indicators = { version: '1.0', /* methods */ };
```

---

3. 10-Step Workflow Implementation

3.1 Workflow Overview

Each step has validation gates preventing forward progress until requirements are met.

3.2 Step Definitions

Step 1: Investment Philosophy ✅ COMPLETE

- Purpose: Acknowledge HCP investment framework
- Requirements: User must check acknowledgment box
- Validation: `state.philosophyAcknowledged = true`
- Status: Fully implemented

Step 2: Data Import & Edit ✅ COMPLETE

- Purpose: Import data and allow manual overrides
- Requirements: Monthly data file or sample generation
- Features:
  - File upload for initialization and monthly data
  - 5 sample data scenarios (Tech Boom, USD Strength, P/E Reversion, International, Mixed)
  - Complete data editing with modal system
  - Manual override tracking with yellow highlighting
- Status: Fully implemented with embedded modules

Step 3: Theme Analysis ✅ COMPLETE

- Purpose: Calculate theme probabilities using IPS v3.10 methodology
- Features:
  - Real indicator-based calculations (not random)
  - 13 indicators across 4 themes with three-tier weighting
  - 16-scenario probability matrix generation
  - Enhanced momentum calculations
- Status: Fully implemented with embedded ThemeCalculator v2.9

Step 4: Scenario Analysis 🔄 IN DEVELOPMENT

- Purpose: Detailed analysis of 16 possible macro scenarios
- Requirements: Theme probabilities from Step 3
- Planned Features:
  - Scenario ranking and probability distribution
  - Risk-return profiles per scenario
  - Correlation analysis between scenarios

Step 5: Portfolio Optimization 🔄 IN DEVELOPMENT

- Purpose: Mean-variance optimization across scenarios
- Requirements: Scenario analysis from Step 4
- Planned Features:
  - Probability-weighted optimization
  - Risk constraints and bounds
  - Asset allocation recommendations

Step 6: Current Positions 🔄 IN DEVELOPMENT

- Purpose: Input current portfolio holdings
- Requirements: User manual input
- Planned Features:
  - Portfolio position entry interface
  - Current allocation analysis
  - Drift calculation from optimal

Step 7: Rebalancing Trades 🔄 IN DEVELOPMENT

- Purpose: Generate specific trades to reach optimal allocation
- Requirements: Current positions and optimal allocation
- Planned Features:
  - Trade list generation
  - Tax optimization considerations
  - Execution priority ranking

Step 8: History 🔄 IN DEVELOPMENT

- Purpose: Historical tracking and audit trail
- Requirements: Previous tracker usage
- Planned Features:
  - Change log and decision history
  - Performance attribution
  - Scenario accuracy tracking

Step 9: Report 🔄 IN DEVELOPMENT

- Purpose: Generate comprehensive analysis report
- Requirements: Completed analysis
- Planned Features:
  - PDF report generation
  - Executive summary
  - Detailed methodology appendix

Step 10: Export 🔄 IN DEVELOPMENT

- Purpose: Export data and results
- Requirements: Completed tracker workflow
- Planned Features:
  - CSV exports for trades, indicators, scenarios
  - JSON backup of complete state
  - Integration with external portfolio systems

---

4. Critical Display Specifications

4.1 Scenario Probability Color Coding

The 16-scenario matrix uses a 5-tier color system based on probability ranges:

Probability Range	Color	CSS Class	Hex Code	Description
> 25%	Dark Green	scenario-very-high	Show Image 🟩 #155724	Extremely likely scenarios
10-25%	Light Green	scenario-high	Show Image 🟩 #28a745	Likely scenarios
5-10%	Yellow	scenario-medium	Show Image 🟨 #ffc107	Moderate probability
1-5%	Light Red	scenario-low	Show Image 🟥 #dc3545	Unlikely scenarios
< 1%	Dark Red/Gray	scenario-very-low	Show Image ⬛ #6c757d	Extremely unlikely

4.2 Theme Color Assignments

Fixed theme colors for consistency across all displays:

Theme	Color Name	CSS Class	Hex Code	RGB
USD	Red	theme-usd	Show Image 🟥 #dc3545	rgb(220, 53, 69)
AI/Innovation	Blue	theme-ai	Show Image 🟦 #007bff	rgb(0, 123, 255)
P/E	Yellow	theme-pe	Show Image 🟨 #ffc107	rgb(255, 193, 7)
International	Green	theme-intl	Show Image 🟩 #28a745	rgb(40, 167, 69)

4.3 Data Confidence Indicators

Separate from probability colors, data confidence shows quality of underlying data:

Confidence Level	When to Use	Display
HIGH	All indicators fresh, complete history	Green dot
MEDIUM	Some stale data or interpolation	Yellow dot
LOW	Significant missing data	Red dot

5. Three-Tier Signal Framework

5.1 Fixed Tier Weights

- **Canary:** 35% (early warning)
- **Primary:** 40% (core signals)
- **Structural:** 25% (confirmation)

5.2 Indicator Classification

Tier	Indicators	Weight	Per-Indicator Weight
Canary	DXY, QQQ/SPY, Risk Premium, ACWX/SPY	35%	8.75% each
Primary	Forward P/E, Net Margins, Yuan SWIFT, CAPE	40%	10% each
Structural	Productivity, Reserve Share, Gold Purchases, TIC Flows	25%	6.25% each

---

6. UI Display Requirements

6.1 Step 3 Theme Display

Each theme shows:

- Theme name with theme color bar
- Percentage probability (large, bold)
- NO confidence labels on probabilities
- Theme color fill proportional to probability

6.2 Scenario Matrix Display

- All 16 scenarios in binary order (0000-1111)
- Rank numbers and colors change based on probabilities
- Binary code for each scenario
- Probability percentage with appropriate color coding

6.3 Missing Data Handling

If indicators are missing:

- Display error message listing missing indicators
- Prevent calculation until all data present

- Show which theme is affected

Example error:

```
Missing Innovation indicators:
- qqq_spy: QQQ/SPY Ratio
- net_margins: S&P Net Margins

Please ensure data file contains all 13 indicators.
```

---

7. Data Requirements

7.1 Input Data Structure (FileHandler v1.4 Compatible)

```
javascript
```

```
{
  "usd": {
    "dxy": {
      "current": 104.2,
      "history": [102.5, 103.1, 103.8, 104.0, 104.2]
    },
    "gold_purchases": {
      "current": 29.1,
      "history": [28.5, 28.8, 29.0, 29.1]
    }
    // ... other USD indicators
  },
  "innovation": {
    "qqq_spy": {
      "current": 0.82,
      "history": [0.78, 0.79, 0.80, 0.81, 0.82]
    }
    // ... other innovation indicators
  },
  "valuation": {
    "forward_pe": {
      "current": 21.4,
      "history": [20.8, 21.0, 21.2, 21.4]
    }
    // ... other valuation indicators
  },
  "international": {
    "acwx_spy": {
      "current": 0.93,
      "history": [0.91, 0.92, 0.93]
    }
    // ... other international indicators
  }
}
```

8. Common Implementation Errors to Avoid

8.1 Data Key Mismatches

❌ Wrong: `'qqqSpy'`, `'netMargins'` (camelCase)

✅ Right: `'qqq_spy'`, `'net_margins'` (snake_case)

8.2 Moving Average Calculations

❌ Wrong: Using daily data for monthly indicators
✅ Right: Month-end standardization with proper baselines

8.3 Momentum Calculation Baseline

❌ Wrong: Compare current vs immediate previous
✅ Right: Compare current vs 6 periods historical baseline

---

9. Quality Assurance Framework

9.1 Module Testing Requirements

- **Unit Tests:** Each module tested in isolation
- **Integration Tests:** Module compatibility verification
- **Data Flow Tests:** FileHandler → ThemeCalculator → UI
- **Scenario Tests:** All 5 sample scenarios generate expected results

9.2 Deployment Validation

Before each release:

□ All sample scenarios work correctly
□ Theme probabilities show proper differentiation
□ Data editing modal functions correctly
□ Manual overrides highlight properly (yellow)
□ File size remains under 150KB

---

10. Version Control Strategy

10.1 Module Versioning

Each module maintains independent version numbers:

- FileHandler v1.4
- ThemeCalculator v2.9
- DataEditor v1.0
- Indicators v1.0

10.2 Integration Versioning

Complete system versions:

- v6.4.3: Current production (Steps 1-3 complete)
- v7.0+: Will include Steps 4-6 (Optimization modules)

10.3 Release Process

1. **Development:** Create/update modular .js files
2. **Testing:** Integration tests with script tag loading
3. **Embedding:** Integrate modules into single HTML file
4. **Validation:** Test all embedded functionality
5. **Release:** Deploy integrated HTML with incremented version

---

11. File Size Management

11.1 Current Status

- Base HTML/CSS/Navigation: ~50KB
- All 4 embedded modules: ~62KB
- **Total v6.4.3:** ~112KB

11.2 Projected Growth

- Remaining 6 modules: ~90KB estimated
- **Final system estimate:** ~200KB
- **Industry context:** Very reasonable for modern web apps

11.3 Optimization Options (if needed)

- **Minification:** Could reduce by 30-40%
- **Selective Loading:** Load modules on-demand
- **Module Splitting:** Core vs Advanced versions

---

12. Hybrid Development Architecture

12.1 Architecture Benefits

Development Advantages:

- Clean module separation for maintainability
- Independent testing of each component
- Clear version control per module
- Easier debugging and feature development

Deployment Advantages:

- Single-file deployment (no dependencies)
- Works completely offline
- No external CDN requirements
- Simple hosting and distribution

12.2 Development Workflow

Phase 1: Modular Development

```
/src
├── file_handler_v1_4.js
├── theme_calculator_v2_9.js
├── data_editor_v1_0.js
└── indicators_v1_0.js
```

Phase 2: Integration Testing

```html
<script src="file_handler_v1_4.js"></script>
<script src="theme_calculator_v2_9.js"></script>
<script src="data_editor_v1_0.js"></script>
<script src="indicators_v1_0.js"></script>
```

Phase 3: Embedded Deployment

```html
```

```
<script>
// FileHandler v1.4 embedded
const FileHandler = { /* complete module */ };

// ThemeCalculator v2.9 embedded
const ThemeCalculator = { /* complete module */ };
// ... etc
</script>
```

12.3 Integration Challenges Solved

Module Loading Issues: External references don't work in single-file deployment

- **Solution:** Embed complete modules in HTML

Version Synchronization: Different modules may have compatibility issues

- **Solution:** Test integration before embedding

File Size Growth: Adding modules increases deployment size

- **Solution:** Monitor size, optimize if needed (currently 112KB is very reasonable)

---

13. Critical Troubleshooting Guide

13.1 Integration Bug Patterns

Based on v6.4.1→v6.4.3 development experience, these are critical failure patterns:

Pattern 1: Undefined Variable References

Symptom: All theme probabilities show uniform values (15.0%) **Root Cause:** Variable scope issues during module embedding **Example Bug:**

```javascript
// In embedded function, variable 'themeData' undefined
Object.entries(themeIndicators).forEach(([key, config]) => {
  const indicator = themeData[config.dataKey]; // ❌ themeData undefined
});
```

Fix: Add missing variable definition

```javascript
const themeData = dataIndicators[theme]; // ✅ Proper definition
```

Pattern 2: Momentum Calculation Baseline Errors

Symptom: All theme probabilities show 50.0% (neutral fallback) **Root Cause:** Momentum comparison uses wrong historical baseline **Example Bug:**

```javascript
const previous = indicator.history[indicator.history.length - 1]; // ❌ Same as current
const momentum = (current - previous) / previous; // Always 0
```

Fix: Use proper historical baseline

```javascript
const previous = indicator.history[indicator.history.length - 6]; // ✅ 6 periods back
```

Pattern 3: Data Structure Mismatches

Symptom: Module functions fail to find indicators **Root Cause:** Different data key naming between modules **Prevention:** Standardize on snake_case for all indicators

13.2 Testing Protocol

Pre-Release Validation:

1. **Sample Data Test:** Generate all 5 scenarios, verify different results
2. **Theme Calculation Test:** Verify probabilities show realistic differentiation
3. **Data Editing Test:** Confirm modal system works and highlights overrides
4. **Console Logging:** Check for calculation step-by-step verification

Expected Tech Boom Results:

- AI Productivity Boom: 70-80%
- USD Dominance Decline: 20-30%
- P/E Mean Reversion: 30-40%
- International Outperformance: 25-35%

Red Flags:

- All themes showing same probability (15%, 50%, etc.)
- Console errors about undefined variables
- Buttons not responding or showing "✅ undefined generated"

13.3 Debug Console Usage

For troubleshooting, enable detailed logging:

```javascript
console.log('=== THEME MOMENTUM CALCULATION ===');
console.log('Data structure themes:', Object.keys(dataIndicators));
console.log('Individual indicator momentum:', momentum);
console.log('Final theme probabilities:', themeProbabilities);
```

13.4 Integration Testing Requirements

Before embedding any new module:

Standalone Testing:

```html
<script src="new_module_v1_0.js"></script>
<script>
// Test module functions independently
console.log('Module loaded:', NewModule.version);
</script>
```

Integration Testing:

```html
<script src="file_handler_v1_4.js"></script>
<script src="new_module_v1_0.js"></script>
<script>
// Test data flow between modules
const data = FileHandler.generateSampleData('tech_boom');
const result = NewModule.processData(data);
console.log('Integration test result:', result);
</script>
```

Only Then Embed: After successful integration testing

13.5 Emergency Rollback Procedures

If a new version has critical bugs:

1. **Immediate:** Revert to last known working version (e.g., v6.4.2 → v6.4.1)
2. **Investigate:** Use standalone module testing to isolate the problem
3. **Fix:** Address the specific integration issue
4. **Validate:** Complete testing protocol before re-release
5. **Document:** Add bug pattern to this troubleshooting guide

---

14. Future Development

14.1 Immediate Priorities (Steps 4-6)

- **Optimizer v1.0:** Mean-variance optimization module
- **PositionManager v1.0:** Portfolio input interface
- **TradeGenerator v1.0:** Rebalancing calculation engine

14.2 Version 7.0 Vision

- Complete 10-step workflow
- PDF report generation
- Advanced optimization constraints
- Real-time data integration capabilities

14.3 Long-term Enhancements

- **Cloud Integration:** Optional cloud sync and sharing
- **Multi-Portfolio Support:** Manage multiple strategies
- **Performance Attribution:** Track strategy effectiveness
- **Mobile Optimization:** Responsive design improvements

---

15. Implementation Status Summary

15.1 Current Capabilities (v6.4.3)

✅ **Fully Functional Steps 1-3:**

- Complete investment philosophy workflow
- Real sample data generation (5 scenarios)
- Professional data editing with manual overrides
- Actual theme analysis using IPS v3.10 calculations
- 16-scenario probability matrix generation

15.2 Technical Achievements

- **Single-file deployment** with embedded modules
- **112KB total size** for production-ready system
- **Offline capability** with full functionality
- **Hybrid architecture** balancing development and deployment needs
- **Comprehensive troubleshooting** documentation

15.3 Quality Validation

- All integration bugs identified and resolved
- Sample scenarios generate expected probability differentiation
- Data editing system fully functional with override tracking
- Theme calculations use proper momentum baselines
- Complete troubleshooting guide prevents regression

---

16. Appendices

Appendix A: Complete Indicator List (13 Total)

Theme	Indicators	Data Keys
USD	DXY, Gold Purchases, Yuan SWIFT, Reserve Share	`dxy`, `gold_purchases`, `yuan_swift`, `reserve_share`
Innovation	QQQ/SPY, Productivity, Net Margins	`qqq_spy`, `productivity`, `net_margins`
Valuation	Risk Premium, Forward P/E, CAPE	`risk_premium`, `forward_pe`, `cape`
International	ACWX/SPY, S&P vs World, US % ACWI, TIC Flows	`acwx_spy`, `sp_vs_world`, `us_pct_acwi`, `tic_flows`

Appendix B: Version History

- **v3.4.3** (2025-08-31): Added comprehensive troubleshooting guide and integration bug documentation

- **v3.4.2** (2025-08-31): Added hybrid development architecture documentation
- **v3.1** (2025-08-25): IPS 3.8 three-tier framework, 13 indicators
- **v3.0** (2025-08-25): Enhanced probability framework with dual-file architecture
- **v2.3** (2025-08-25): Momentum analysis requirements update
- **v2.2** (2025-08-24): Aligned with v5.6 implementation
- **v2.1** (2025-08-24): Data Collector v3.7+ compatibility
- **v2.0** (2025-08-23): Initial PRD

Appendix C: Integration Test Cases

Test Case 1: Tech Boom Scenario

- Input: FileHandler.generateSampleData('tech_boom')
- Expected: AI theme 70-80%, USD theme 20-30%
- Validates: Proper momentum calculation and theme differentiation

Test Case 2: Data Editor Override

- Input: Manual edit of QQQ/SPY ratio
- Expected: Yellow highlighting, original value preserved
- Validates: Override system and audit trail

Test Case 3: Theme Calculation Chain

- Input: Complete 13-indicator dataset
- Expected: 16-scenario matrix with realistic probabilities
- Validates: Full calculation pipeline functionality

End of Product Requirements Document v3.4.3

Document Control:

- **Filename:** hcp_tracker_prd_v3_4_3.md
- **Last Updated:** 2025-08-31 23:45:00 UTC
- **Next Update:** Upon completion of Step 4 development
- **Approval:** Ready for v6.4.3 production deployment