Oracle EPM Predictive Cash Forecasting

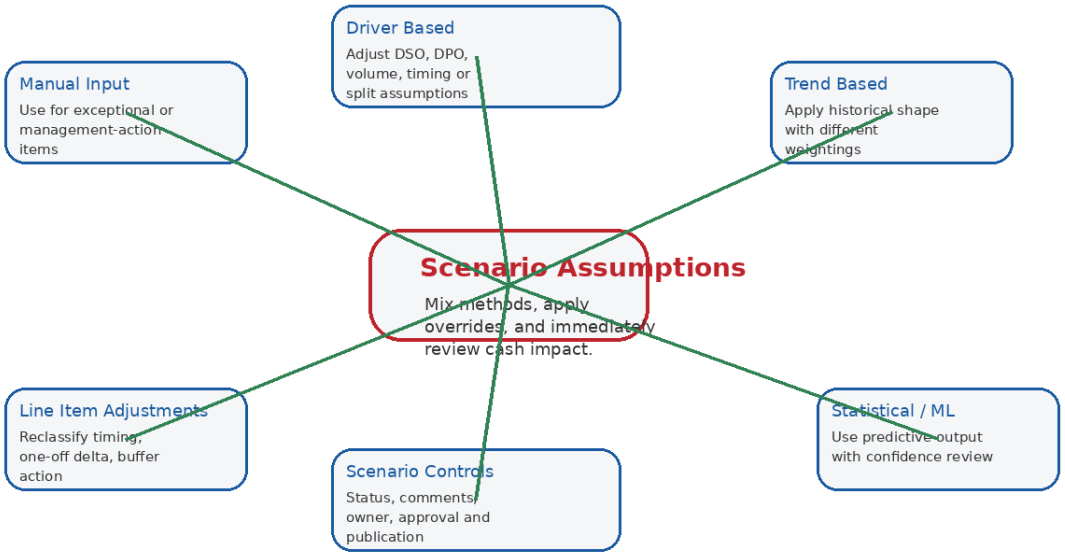
What-if Scenarios

Advanced Lab Guide | Professional BISP Edition

Product Design Documentation
Implementation Guide
Training Material
Technical Reference
Troubleshooting Handbook

SmartArt: What-if Scenario Building Blocks

blend methods by line item or period



Oracle-inspired layout | White enterprise background | Red accent elements | Dark charcoal headings

1. Introduction and Learning Objectives

This lab guide explains how to design, configure, test, compare, and operationalize What-if Scenarios in Oracle EPM Predictive Cash Forecasting (PCF). The guide is intended for advanced-stage students, implementation consultants, treasury analysts, and product owners who need a rigorous, Oracle-inspired reference for scenario analysis.

- Understand where What-if Scenarios fit within the PCF operating model.
- Configure scenario assumptions by line item, period, and forecast method.
- Blend manual, driver-based, trend-based, statistical, and predictive approaches inside a scenario.
- Compare baseline and alternative outcomes and decide which scenario should be promoted.
- Use validation, audit, approval, and troubleshooting practices suitable for production use.

Why What-if Scenarios matter

What-if analysis allows treasury and finance teams to test potential business conditions—such as slower collections, higher supplier payments, one-time tax outflows, delayed capex, or additional financing—without overwriting the baseline rolling forecast. It provides a controlled environment for assumption changes, impact evaluation, and management decision support.

2. Product Design View

In the Oracle EPM PCF context, What-if Scenarios are best thought of as governed forecast variants that reuse the rolling forecast structure but allow selective method changes, assumption overrides, and manual interventions. A scenario should preserve traceability to a baseline and keep enough metadata to support audit, review, and publication.

Design principles

- Baseline first: every scenario should start from a known, approved baseline.
- Targeted recalculation: reprocess only impacted line items where possible for speed.
- Line-item flexibility: use different methods for different cash line items and periods.
- Separation of duties: preparer, reviewer, and approver should be distinguishable.
- Explainability: each override should carry rationale, comments, and ownership.
- Comparability: scenario outputs should be easy to compare to baseline and to other scenarios.

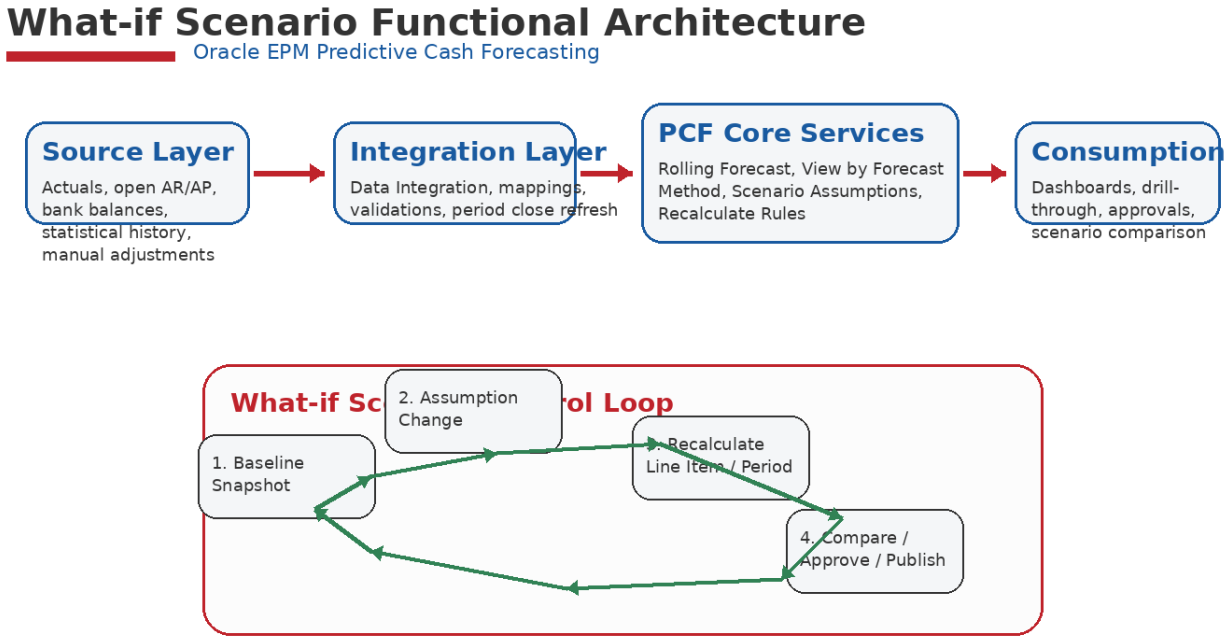


Figure 1. Functional architecture for What-if Scenarios.

3. Functional Architecture and Data Flow

A typical PCF scenario solution combines baseline actuals and forecast data, integration or direct form entry, calculation logic, method selection, scenario storage, dashboard comparison, and approval/publishing controls.

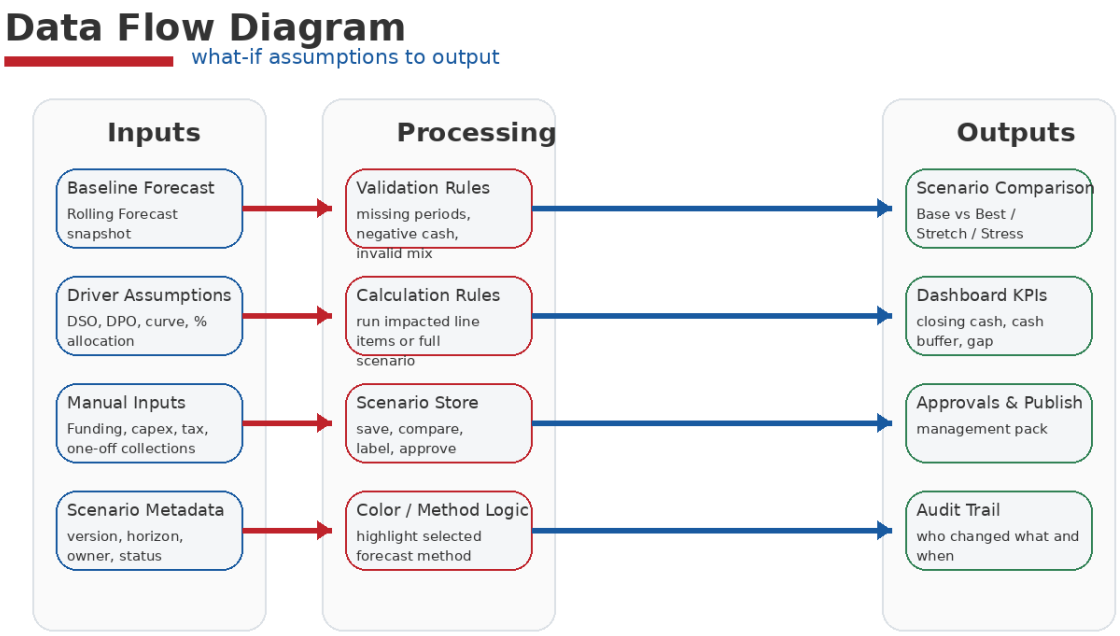


Figure 2. Data flow from baseline forecast and assumptions into scenario output.

Key artifacts in scope

Artifact	Purpose	Typical Owner
Rolling Forecast	Provides the starting baseline and final output view	Treasury / Cash Manager
View by Forecast Method	Explains how each line item is being forecast	Cash Manager
Scenario Assumptions Form	Holds method overrides, weights, and comments	Treasury Analyst
Manual Input Form	Captures one-off or management-action values	Treasury Analyst
Compare Dashboard	Quantifies deltas between scenarios and baseline	Management / Treasury
Audit & Logs	Tracks changes, rule execution, validation issues	Administrator / Support

4. Method Selection and Scenario Logic

A scenario can change the forecast method of a line item entirely, blend methods, or keep the base method and only modify its assumptions. In practice, the finance team should decide whether the change represents a structural shift, a one-time event, or a temporary sensitivity.

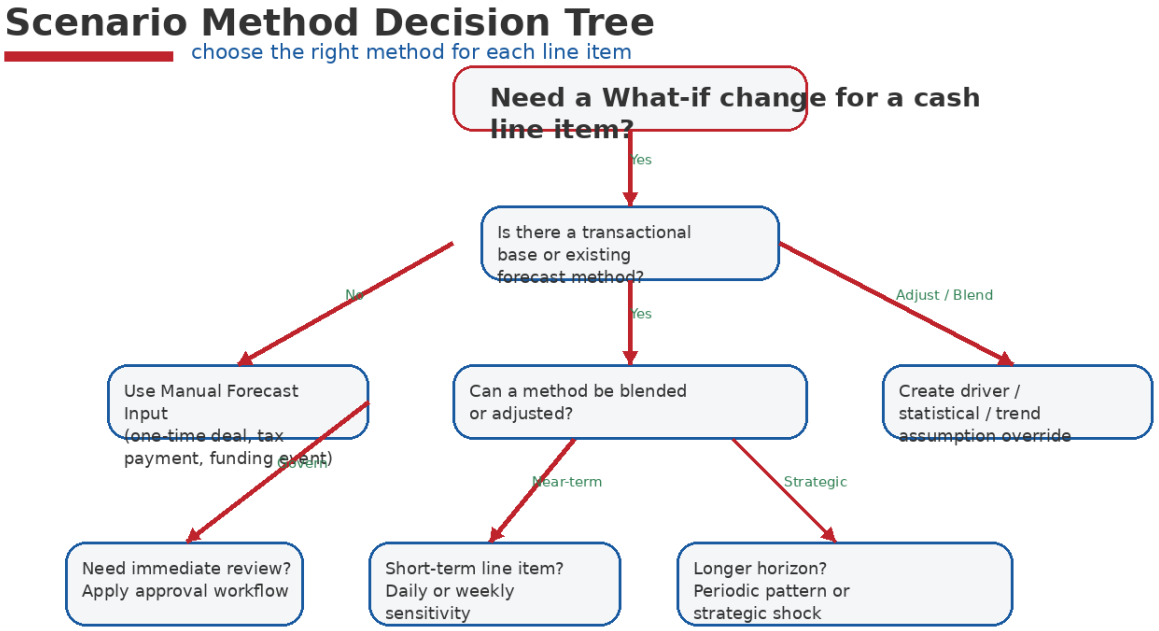


Figure 3. Decision tree for choosing the best scenario treatment.

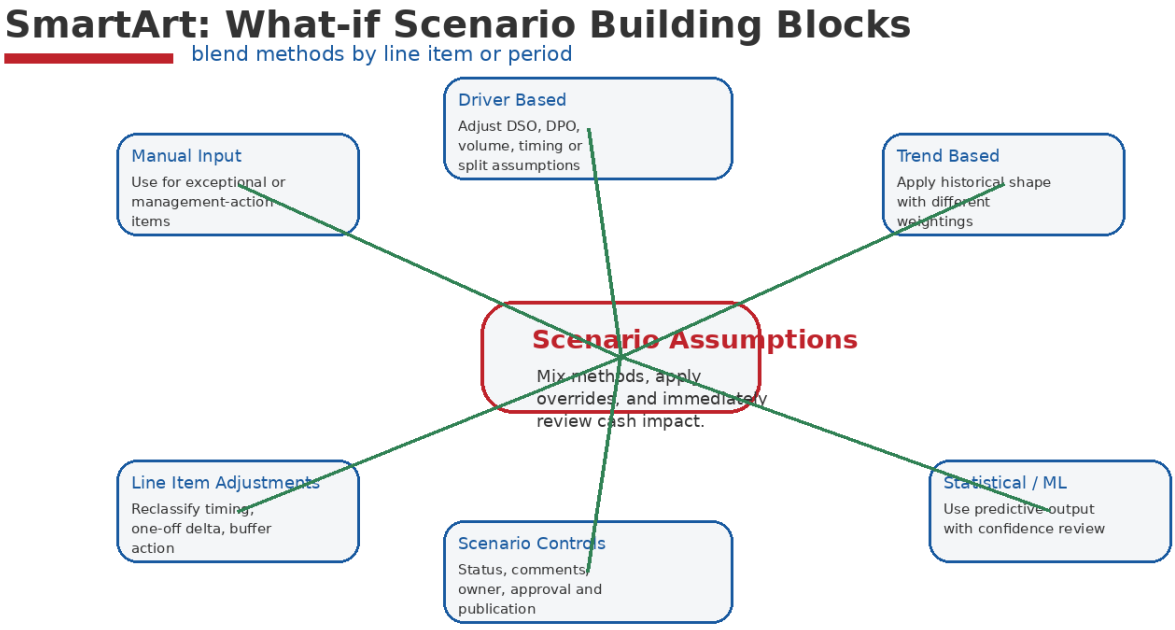


Figure 4. SmartArt-style view of scenario building blocks.

Typical scenario use cases

- Collections stress scenario: reduce collection percentages or delay customer receipts in selected weeks.
- Supplier pressure scenario: accelerate payments or reduce negotiated DPO improvement.
- Tax / regulatory event: add large manual outflows in the relevant period.
- Capex deferral scenario: postpone non-essential investing outflows to preserve liquidity.
- Funding action scenario: add borrowing, equity inflow, or intercompany funding to close gaps.
- Board review scenario: compare Base, Best Case, Stress Case, and Mitigation Case side by side.

5. Process Flow and Operating Model

The operational flow should be consistent and repeatable. A well-run scenario process enables planners to create, test, validate, analyze, approve, and publish scenarios without creating confusion in the live forecast.

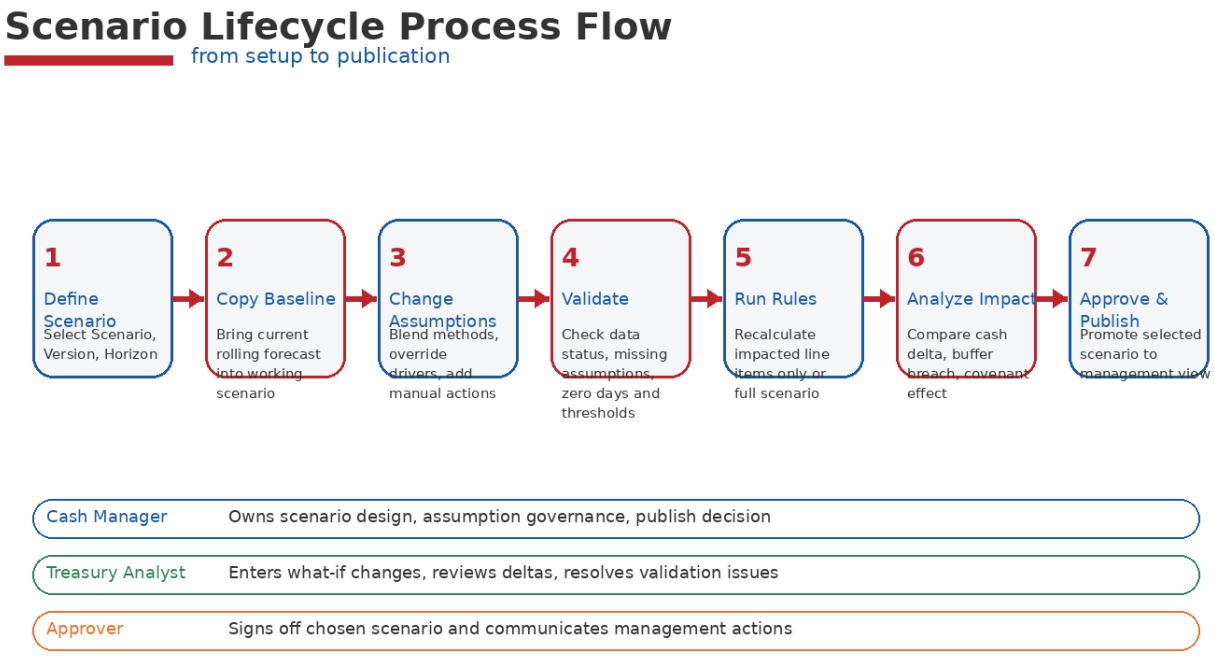


Figure 5. End-to-end scenario lifecycle process flow.

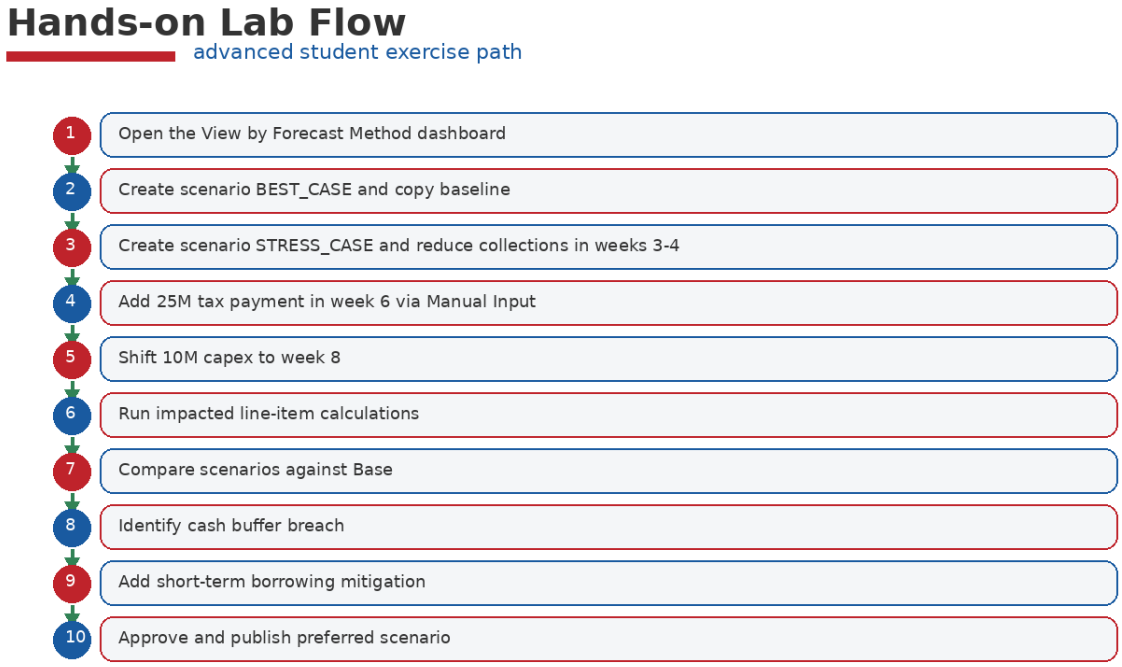


Figure 6. Guided lab path for advanced students.

Recommended roles

Role	Responsibilities
Cash Manager	Own scenario calendar, approve design standards, select publishable scenario
Treasury Analyst	Prepare scenarios, edit assumptions, run validation and calculation rules
Administrator	Manage metadata, security, integrations, dimensions, and troubleshooting
Approver / CFO Delegate	Review impact and sign off on management use
Business Stakeholder	Provide operational assumptions such as delayed receipts or deferred spend

6. Configuration Mockup and Dashboard Mockup

The following mockups show a practical Oracle-inspired design for a scenario setup screen and a comparison cockpit. These are not screenshots from the product but are suitable for implementation workshops, design reviews, and training.

Configuration Mockup

Scenario setup form

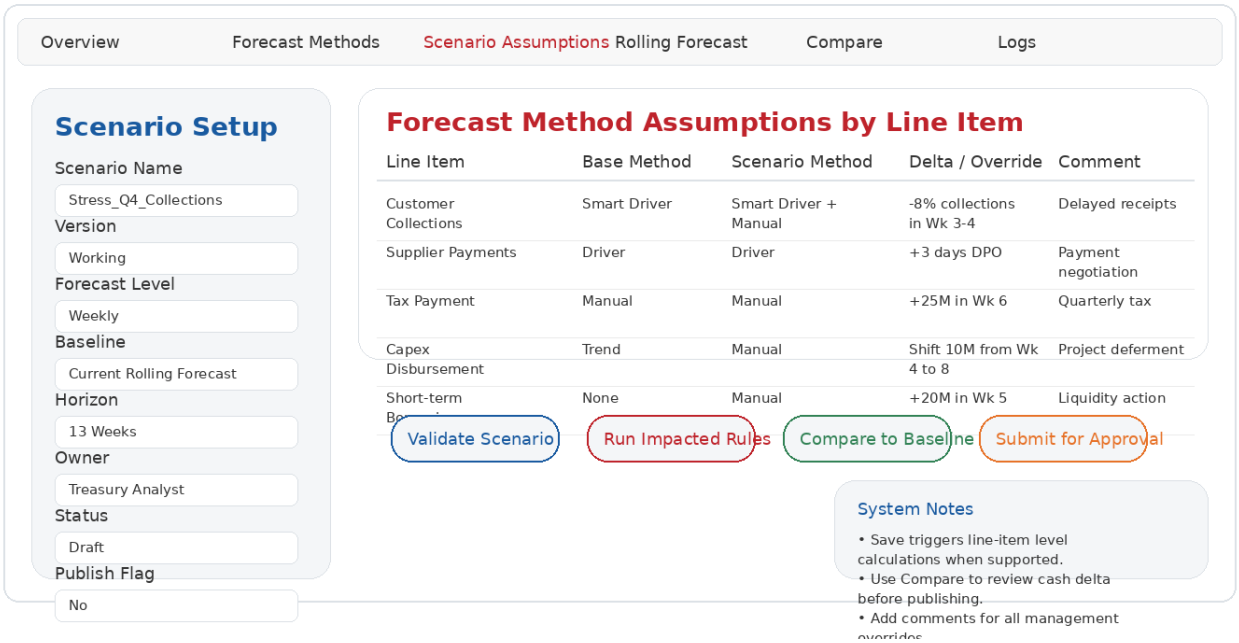


Figure 7. Configuration mockup for scenario assumptions.

Dashboard Mockup

What-if scenario comparison cockpit



Figure 8. Dashboard mockup for scenario comparison and action tracking.

7. Data Model, Dimensions, and Governance

Minimum data concepts

Data Area	Examples	Notes
Scenario Metadata	Scenario Name, Owner, Status, Horizon, Approval Status	Needed for control and audit
Baseline Values	Opening Cash, baseline inflows/outflows	Copied from current rolling forecast
Method Assumptions	DSO, DPO, trend %, predictive weight, manual values	Can differ by line item and period
Comparison Measures	Absolute delta, percentage delta, buffer variance	Required for management analysis
Comments / Audit	Reason for override, ticket ID, sign-off note	Important for governance

Governance controls

- Use a naming standard such as BASELINE_Q4, BEST_Q4, STRESS_Q4, and MITIGATION_Q4.
- Require comments for any manual override or scenario publication.
- Lock published scenarios and keep working scenarios editable.
- Retain only the active scenarios necessary for decision making to prevent user confusion.
- Store a baseline snapshot before major scenario testing cycles.

8. Worked Example – Advanced 13-Week What-if Lab

This lab demonstrates how a treasury analyst can test a stress scenario and a mitigation scenario for a 13-week cash horizon. The example is intentionally detailed so learners can reproduce the logic in a sandbox or a training instance.

Baseline business situation

Metric	Baseline Value
Opening Cash	150.0M
Cash Buffer Requirement	20.0M
Expected Customer Collections (13 weeks)	310.0M
Supplier Payments	205.0M
Payroll & Operating Outflows	78.0M
Tax Payments	12.0M
Capex	22.0M

Scenario A – Stress Case assumptions

- Customer collections in Weeks 3 and 4 are delayed, reducing short-term inflow by 24.0M.
- A 25.0M tax payment occurs in Week 6 instead of later in the quarter.
- Supplier payment pressure accelerates 8.0M of outflow into Week 5.
- No immediate financing action is taken in the first run.

Scenario B – Mitigation Case assumptions

- The organization arranges a 20.0M short-term borrowing draw in Week 5.
- 10.0M of discretionary capex is deferred from Week 4 to Week 8.
- Collections team recovers 8.0M of the delayed receipts by Week 7.

Illustrative week-by-week output

Week	Baseline Closing Cash	Stress Case Closing Cash	Mitigation Case Closing Cash	Comment
1	152.0	152.0	152.0	No change
2	149.0	149.0	149.0	No change
3	155.0	142.0	142.0	Collections delay begins
4	160.0	136.0	136.0	Receipts pushed out
5	158.0	130.0	150.0	Supplier acceleration; borrowing added in mitigation
6	162.0	126.0	146.0	Tax payment hits
7	165.0	129.0	157.0	Partial collection recovery
8	170.0	134.0	172.0	Capex deferment improves cash
9	168.0	140.0	176.0	Gap continues closing
10	172.0	145.0	180.0	Management scenario stabilizes
11	174.0	148.0	183.0	Positive position maintained
12	176.0	152.0	186.0	Near close
13	180.0	142.5	182.0	Stress remains below optimal finish

Interpretation

The Stress Case falls below the required cash buffer in Weeks 5–7 and ends significantly below the baseline. The Mitigation Case restores sufficient liquidity by combining financing and timing actions. In a management review, the Mitigation Case would likely be the preferred publishable scenario while the Stress Case remains a reference risk view.

9. Training Material – Step-by-Step Lab Instructions

Exercise 1: Create scenarios

- Open the View by Forecast Method dashboard and review the baseline rolling forecast.
- Create two scenarios: STRESS_Q4 and MITIGATION_Q4.
- Copy the current rolling forecast into both scenarios.

- Verify that horizon, version, entity, and currency are correct.

Exercise 2: Apply assumption changes

- In STRESS_Q4, reduce collection performance in Weeks 3–4.
- Add a manual tax payment in Week 6.
- Accelerate a supplier payment wave into Week 5.
- Save the form and run impacted line-item rules.

Exercise 3: Add mitigation actions

- In MITIGATION_Q4, start from STRESS_Q4 logic or reapply the same stress assumptions.
- Add a 20.0M funding inflow in Week 5.
- Move 10.0M capex from Week 4 to Week 8.
- Recover 8.0M delayed customer receipts by Week 7.
- Save and rerun scenario calculations.

Exercise 4: Compare and publish

- Open the Compare dashboard and review closing cash, minimum cash buffer, and variance drivers.
- Record findings in scenario comments.
- Submit the preferred scenario for approval.
- Publish only after approver sign-off.

10. Technical Reference

Recommended calculation sequence

Step	Action	Why it matters
1	Validate input completeness	Prevents rule failures and misleading output
2	Save assumption forms	Commits scenario changes
3	Run impacted line-item calculations	Improves performance for scenario testing
4	Refresh compare dashboard	Ensures visuals use latest scenario results
5	Review logs and exceptions	Confirms successful processing
6	Approve and publish	Locks official view

Performance considerations

- Prefer line-item or targeted recalculation over full-scenario recalculation when available.
- Archive obsolete scenarios periodically.
- Use clean naming and comments to help administrators troubleshoot faster.
- Keep scenario horizon aligned with business need—do not overbuild long test cases unnecessarily.

11. Log / Error Examples and Troubleshooting

Scenario design is powerful, but it introduces risk if assumptions are incomplete or poorly governed. The following visual shows realistic warning and error patterns and how they should be interpreted during support activities.

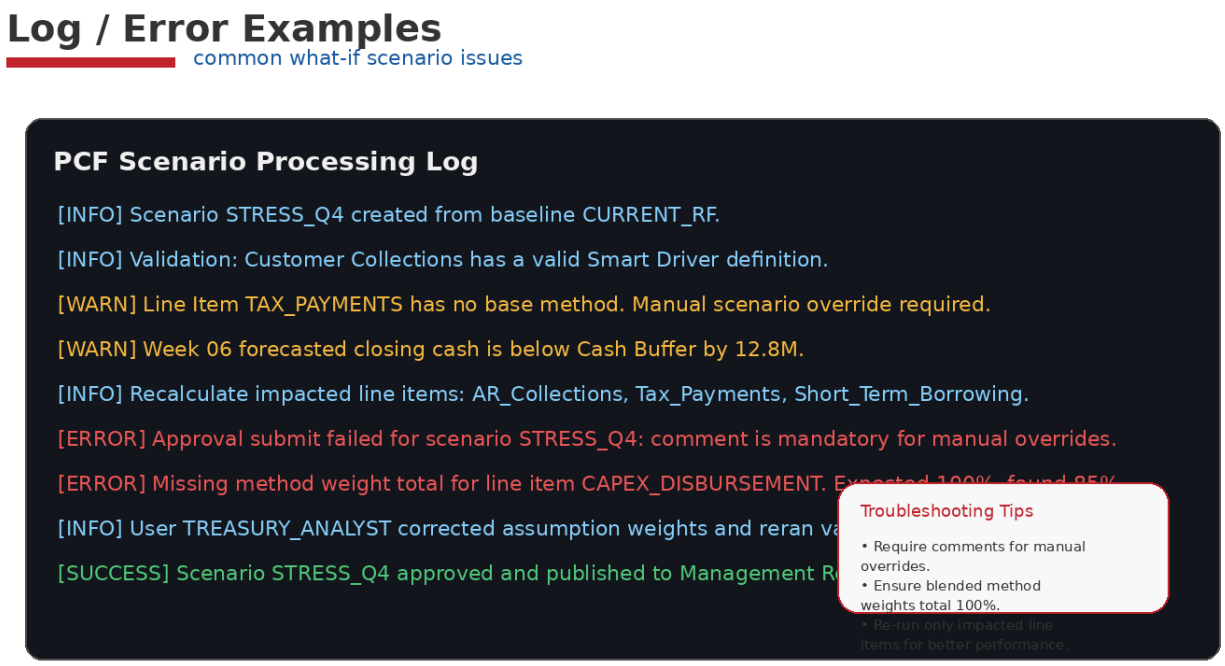


Figure 9. Example processing log and scenario troubleshooting guidance.

Common issues and resolutions

Issue	Likely Cause	Resolution
Scenario not recalculating	Save event or rule was not triggered	Save the form again and rerun the impacted rule
Cash delta looks wrong	Baseline was not copied correctly or wrong scenario selected	Verify scenario metadata and baseline source
Manual override rejected	Comment or required attribute missing	Complete mandatory fields and resubmit
Blended method fails validation	Weights do not total 100%	Correct method allocation percentages
Unexpected negative cash	Scenario assumptions too severe or missing mitigation	Review weeks with breaches and add actions
Dashboard not updated	Comparison layer not refreshed	Refresh form or rerun comparison process

12. Implementation Guide

Suggested delivery phases

- Phase 1 – Design: identify use cases, line items, roles, and scenario governance.
- Phase 2 – Build: configure dimensions, forms, rules, comparison views, and security.
- Phase 3 – Test: execute baseline and scenario test packs, including negative tests.
- Phase 4 – Train: run advanced lab exercises with treasury and finance teams.
- Phase 5 – Go-live: enable scenario approval and controlled publishing.
- Phase 6 – Operate: monitor usage, logs, and feedback for continuous improvement.

Implementation checklist

☐ Scenario naming convention approved

☐ Baseline copy process tested

☐ Method assumption forms validated

☐ Manual input form available for one-off actions

☐ Comparison dashboard configured

☐ Approval workflow agreed

☐ Support runbook and troubleshooting guide prepared

☐ User training completed

☐ Production smoke test executed

13. Conclusion

What-if Scenarios are one of the most valuable capabilities in Predictive Cash Forecasting because they connect treasury insight with management action. When implemented professionally—with strong governance, line-item flexibility, comparison analytics, and clean publish controls—they turn the rolling forecast from a passive report into a decision-support system.

This BISP edition is intentionally structured to serve multiple purposes at once: a design document for consultants, an implementation guide for project teams, a training workbook for advanced learners, a technical reference for administrators, and a troubleshooting handbook for support personnel.