CLAUDE COWORK

Finance & Accounting Plugin Handbook

The Complete Guide for Controllers, Accounting Managers, and Finance Directors

VERSION	1.0.0
PLUGIN	finance

RationalEyes.ai
contact@rationaleyes.ai

February 2026

Table of Contents

The Complete Guide for Controllers, Accounting Managers, and Finance Directors

1. Overview and Purpose

2. Plugin Architecture and Components

3. Command Reference

4. Skills Deep Dive

5. Integration and MCP Connectors

6. Workflows and Use Cases

7. Customization and Extension

The Complete Guide for Controllers, Accounting Managers, and Finance Directors

1. Overview and Purpose

The finance and accounting plugin is designed to streamline the critical workflows that keep your books clean and your financial reporting accurate. From journal entry preparation and account reconciliation to month-end close management, variance analysis, and SOX audit support, this plugin serves as a specialized assistant for accounting and finance professionals.

Important Legal Notice: This plugin assists with finance and accounting workflows but does not provide financial, tax, or audit advice. All outputs should be reviewed by qualified financial professionals before use in financial reporting, regulatory filings, or audit documentation.

Who This Plugin Serves

ROLE	PRIMARY USE CASES
Staff Accountant	Journal entry preparation, account reconciliation, prepaid/accrual schedules
Senior Accountant	Month-end close coordination, variance analysis, reconciliation review
Accounting Manager	Close management, financial statement preparation, process improvement
Controller	Financial reporting, flux analysis, audit coordination, internal controls
Finance Director / VP Finance	Strategic variance analysis, board reporting, SOX compliance oversight
Internal Auditor	Control testing, sample selection, deficiency assessment

What Problems This Plugin Solves

Speed up the month-end close. Automate recurring journal entries, generate reconciliation templates, identify outstanding items, and flag variances that need investigation. What used to take 7-10 days can compress to 3-5 days with systematic automation.

Improve accuracy and consistency. Standardized templates and checklists reduce errors. Every journal entry follows the same format, every reconciliation includes the same review steps, every variance gets the same level of analysis.

Strengthen audit readiness. SOX testing workpapers, sample selections, control documentation, and deficiency classification all follow industry standards. Your audit documentation is consistent and complete.

Reduce manual effort on repetitive tasks. Generating depreciation schedules, prepaid amortization tables, payroll accruals, and standard reconciliations becomes a matter of providing the source data and letting the plugin apply the methodology.

Surface insights faster. Variance analysis decomposes changes into drivers (volume, price, mix, timing) with narrative explanations. You spend less time crunching numbers and more time understanding what changed and why.

Core Capabilities at a Glance

CAPABILITY	WHAT IT DOES	TIME SAVED
Journal Entry Preparation	Generate accruals, depreciation, prepaid amortization, payroll, and revenue entries with supporting detail	60-70% reduction in manual entry prep time
Account Reconciliation	Reconcile GL to subledger, bank, or third-party balances; categorize reconciling items	50-60% faster reconciliation cycle
Income Statement Generation	Produce P&L with period-over-period comparison and variance flagging	Immediate multi-period comparison
Variance Analysis	Decompose variances into drivers with waterfall analysis and narrative explanations	70% reduction in flux analysis time
SOX Testing Support	Generate sample selections, testing workpapers, and control assessments	50% faster test workpaper preparation
Close Management	Track close tasks, dependencies, and status across the team	Visibility into close progress in real-time

Installation and Setup

Cowork (recommended): Open **Plugin Settings** in the Cowork desktop app, find **Finance**, and click **Install**. The plugin activates immediately — no CLI required.

Claude Code CLI (alternative): If you are using Claude Code in the terminal rather than Cowork, install via:

```
claude plugins add knowledge-work-plugins/finance
```

> **Note:** *All standard Cowork plugins, including Finance, are available from Plugin Settings with a single click. The CLI command above is only needed for Claude Code terminal users.*

After installation, connect your financial data sources via MCP servers for the best experience. The plugin works without integrations (you can paste data or upload files), but connecting your ERP, data warehouse, and communication tools enables automatic data retrieval and streamlined workflows.

2. Plugin Architecture and Components

The finance plugin follows the standard Cowork plugin structure. Understanding this architecture helps you extend, customize, and troubleshoot the plugin effectively.

Directory Structure

```
finance/
├── .claude-plugin/
│   └── plugin.json             # Plugin identity and metadata
├── commands/
│   ├── journal-entry.md        # /journal-entry command
│   ├── reconciliation.md       # /reconciliation command
│   ├── income-statement.md     # /income-statement command
│   ├── variance-analysis.md    # /variance-analysis command
│   └── sox-testing.md          # /sox-testing command
├── skills/
│   ├── journal-entry-prep/
│   │   └── SKILL.md            # JE preparation methodology
│   ├── reconciliation/
│   │   └── SKILL.md            # Reconciliation best practices
│   ├── financial-statements/
│   │   └── SKILL.md            # Statement formats and GAAP guidance
│   ├── variance-analysis/
│   │   └── SKILL.md            # Variance decomposition techniques
│   ├── close-management/
│   │   └── SKILL.md            # Month-end close workflows
│   └── audit-support/
│       └── SKILL.md            # SOX testing methodology
├── .mcp.json                   # MCP server connections
├── CONNECTORS.md               # Tool category documentation
├── README.md                   # Plugin documentation
└── LICENSE                     # License information
```

Component Breakdown

Commands (user-invoked actions):

- `/journal-entry` — Prepare specific journal entry types
- `/reconciliation` — Reconcile specific accounts
- `/income-statement` — Generate financial statements
- `/variance-analysis` — Analyze variances by driver
- `/sox-testing` — Generate SOX testing workpapers

Skills (automatically-loaded domain knowledge):

- `journal-entry-prep` — Entry types, documentation requirements, review checklist
- `reconciliation` — Reconciliation methodology, aging, escalation thresholds
- `financial-statements` — GAAP presentation, common adjustments, flux methodology

- `variance-analysis` — Decomposition techniques, materiality thresholds, narrative generation
- `close-management` — Close calendar, task dependencies, status tracking
- `audit-support` — SOX 404 testing methodology, sample selection, deficiency classification

MCP Integrations (external tool connections):

- Data warehouse (Snowflake, Databricks, BigQuery) — query financial data
- Email (Microsoft 365) — send reports, request approvals
- Office suite (Microsoft 365) — generate Excel workpapers
- Chat (Slack) — close status updates
- ERP / Accounting system (placeholder — no standard MCP yet) — pull trial balance, subledger data

The Placeholder System

The finance plugin uses the `~~category` placeholder system to remain tool-agnostic. References like `~~data warehouse` or `~~ERP` in command and skill files get replaced with your organization's specific tools during customization.

Placeholder categories in this plugin:

CATEGORY	PLACEHOLDER	PRE-CONFIGURED OPTIONS	OTHER ALTERNATIVES
Data warehouse	`~~data warehouse`	Snowflake, Databricks, BigQuery	Redshift, PostgreSQL
Email	`~~email`	Microsoft 365	Gmail (via Google Workspace)
Office suite	`~~office suite`	Microsoft 365	Google Workspace
Chat	`~~chat`	Slack	Microsoft Teams
ERP / Accounting	`~~erp`	None (not yet available)	NetSuite, SAP, QuickBooks, Xero
Analytics / BI	`~~analytics`	None (not yet available)	Tableau, Looker, Power BI

For detailed connector information, see `CONNECTORS.md` in the plugin directory.

3. Command Reference

Commands are slash-invoked actions that perform specific finance and accounting tasks. Each command follows a structured workflow to ensure completeness and accuracy.

/journal-entry

Purpose: Prepare journal entries with proper debits, credits, and supporting documentation.

Usage:

```
/journal-entry <type> <period>
```

Arguments:

- `type` — The entry type to prepare
    - `ap-accrual` — Accounts payable accruals for goods/services received but not invoiced
    - `fixed-assets` — Depreciation and amortization
    - `prepaid` — Prepaid expense amortization
    - `payroll` — Payroll accruals (salaries, benefits, taxes, bonuses)
    - `revenue` — Revenue recognition entries
- `period` — Accounting period (e.g., `2024-12`, `2024-Q4`, `2024`)

Examples:

```
/journal-entry ap-accrual 2024-12
/journal-entry fixed-assets 2024-Q4
/journal-entry prepaid 2024-12
```

Workflow:

1. **Gather source data** — If ~~erp or ~~data warehouse is connected, automatically pull trial balance and subledger detail. Otherwise, prompt for GL balances and supporting schedules.
2. **Calculate the entry** — Apply standard methodology for the entry type (see Section 4.1 for detailed methodologies).
3. **Generate the entry** — Present in standard format with account codes, debits, credits, department coding, and memo.
4. **Review checklist** — Verify debits equal credits, correct period, valid account codes, supporting detail present, reversal flag set appropriately.
5. **Output** — Formatted journal entry, supporting calculations, comparison to prior period, and posting instructions.

Output format:

```
Journal Entry: AP Accrual — December 2024
Prepared by: [User]
Date: 2024-12-31

| Line | Account | Account Name         | Debit  | Credit | Dept | Memo
├──────┼─────────┼─────────────────────┼────────┼────────┼──────┼──────────────────────
| 1    | 6200    | Professional Fees    | 15,000 |        | G&A  | Accrual - Legal fees
| 2    | 2100    | Accrued Liabilities |        | 15,000 | G&A  | Accrual - Legal fees
|      |         | **Total**            | 15,000 | 15,000 |      |

Supporting Detail:
- Based on legal invoice estimate for Q4 services ($15K per counsel email 12/28)
- Reference: Email from outside counsel dated 12/28/2024
- Prior period accrual: $12,500 (Q3 2024)

Reversal: Yes — auto-reverse on 1/1/2025
```

When to use this command:

- Month-end or quarter-end close
- Any time a manual journal entry is needed
- When you need a standardized, audit-ready journal entry template

---

/reconciliation

Purpose: Reconcile GL account balances to subledger, bank statement, or third-party balances.

Usage:

```
/reconciliation <account> <period>
```

Arguments:

- `account` — The account or category to reconcile
  - `cash` or `bank` — Bank reconciliation
  - `accounts-receivable` or `ar` — AR subledger reconciliation
  - `accounts-payable` or `ap` — AP subledger reconciliation
  - `fixed-assets` or `fa` — Fixed asset subledger reconciliation
  - `intercompany` or `ic` — Intercompany balance reconciliation

  - `prepaid` — Prepaid expense schedule reconciliation
  - `accrued-liabilities` — Accrued liabilities detail reconciliation
  - Or any specific GL account code (e.g., `1010`, `2100`)
- `period` — Period end date (e.g., `2024-12-31`, `2024-12`)

Examples:

```
/reconciliation cash 2024-12-31
/reconciliation ar 2024-12
/reconciliation intercompany 2024-Q4
```

Workflow:

1. **Gather both sides** — Pull GL balance and the comparison source (subledger, bank statement, third-party balance).
2. **Compare balances** — Calculate the difference between the two sides.
3. **Identify reconciling items** — Categorize as timing differences (will clear), permanent differences (require adjustment), or items requiring investigation.
4. **Generate reconciliation workpaper** — Standard format showing GL balance, adjustments, reconciling items, and agreement to the other side.
5. **Age outstanding items** — Track how long reconciling items have been outstanding.
6. **Flag items for escalation** — Identify aged items, large items, or growing balances that need management attention.

Output format:

```
ACCOUNT RECONCILIATION
Account: 1010 — Cash - Operating Account
Period End: December 31, 2024
Prepared by: [User]
Date Prepared: January 3, 2025

RECONCILIATION SUMMARY
Balance per General Ledger:              $1,245,823.45

Add: Reconciling items increasing GL
  Deposit in transit - 12/31             $25,000.00
                                         ───────────
  Subtotal additions:                    $25,000.00

Less: Reconciling items decreasing GL
  Outstanding check #5821                ($3,450.00)
  Outstanding check #5829                ($1,275.50)
  Bank service charge - December         ($35.00)
                                         ───────────
  Subtotal deductions:                   ($4,760.50)

Adjusted GL Balance:                     $1,266,062.95
Balance per Bank Statement:              $1,270,823.45

Add: Unrecorded interest income          $100.00
Less: Outstanding checks                 ($4,725.50)
      Bank fees not yet recorded         ($35.00)

Adjusted Bank Balance:                   $1,266,062.95

DIFFERENCE:                              $0.00
```

When to use this command:

- Month-end or quarter-end close
- Any account requiring reconciliation to an external source
- When investigating account balance discrepancies

/income-statement

Purpose: Generate an income statement with period-over-period comparison and variance analysis.

Usage:

```
/income-statement <period-type> <period>
```

Arguments:

- `period-type` — The reporting frequency
  - `monthly` — Single month P&L with prior month and prior year month comparison
  - `quarterly` — Quarter P&L with prior quarter and prior year quarter comparison
  - `annual` — Full year P&L with prior year comparison
  - `ytd` — Year-to-date P&L with prior year YTD comparison
- `period` — The period to report (e.g., `2024-12`, `2024-Q4`, `2024`)

Examples:

```
/income-statement monthly 2024-12
/income-statement quarterly 2024-Q4
/income-statement ytd 2024-12
```

Workflow:

1. **Gather financial data** — Pull income statement data for current and comparison periods.
2. **Generate income statement** — Present in multi-column format with revenue, COGS, gross profit, operating expenses, and net income.
3. **Calculate variances** — Compute dollar and percentage variances for each line item.
4. **Flag material variances** — Apply materiality thresholds and flag items requiring investigation.
5. **Generate key metrics** — Calculate margins (gross, operating, net) and show period-over-period changes.
6. **Summarize material variances** — List all flagged variances with preliminary drivers.

Output includes:

- Multi-column income statement (current, prior, variance $, variance %)
- Key metrics summary (margins, growth rates)
- Material variance listing
- Suggested follow-up questions for unexplained variances

When to use this command:

- Monthly or quarterly financial reporting
- Board or investor reporting packages
- Management review meetings

- Identifying areas for deeper variance analysis (use `/variance-analysis` for drill-down)

/variance-analysis

Purpose: Decompose variances into underlying drivers with narrative explanations and waterfall analysis.

Usage:

```
/variance-analysis <area> <period-comparison>
```

Arguments:

- `area` — The area to analyze
  - `revenue` — Revenue variance by stream, product, geography
  - `opex` — Operating expense variance by category, department
  - `capex` — Capital expenditure variance vs budget
  - `headcount` — Headcount and compensation variance
  - `cogs` or `cost-of-revenue` — Cost of revenue variance
  - `gross-margin` — Gross margin with mix and rate effects
  - Any specific GL account or account group
- `period-comparison` — The periods to compare
  - `2024-12 vs 2024-11` — Month over month
  - `2024-12 vs 2023-12` — Year over year
  - `2024-Q4 vs 2024-Q3` — Quarter over quarter
  - `2024-12 vs budget` — Actual vs budget
  - `2024-12 vs forecast` — Actual vs latest forecast

Examples:

```
/variance-analysis revenue 2024-Q4 vs 2024-Q3
/variance-analysis opex 2024-12 vs budget
/variance-analysis gross-margin 2024-12 vs 2023-12
```

Workflow:

1. **Gather data** — Pull actuals for both comparison periods and supporting operational metrics.
2. **Calculate top-level variance** — Total dollar and percentage variance.

3. **Decompose by driver** — Break down the variance using appropriate methodology (price/volume for revenue, headcount/rate for compensation, etc.).
4. **Generate waterfall** — Show how each driver contributes to the total variance.
5. **Write narrative explanations** — For each significant driver, provide 2-3 sentence explanation of the business reason.
6. **Identify unexplained variances** — Flag residual amounts that need further investigation.

Output includes:

- Top-level variance summary
- Waterfall analysis (text or chart format)
- Driver-by-driver breakdown with percentages
- Narrative explanations for each material driver
- Unexplained variance flag with investigation suggestions

When to use this command:

- Following up on material variances flagged in `/income-statement`
- Budget variance explanations for management or board
- Preparing variance commentary for earnings releases or investor updates
- Root cause analysis for unexpected results

/sox-testing

Purpose: Generate SOX sample selections, testing workpapers, and control assessments.

Usage:

```
/sox-testing <control-area> <period>
```

Arguments:

- `control-area` — The control area to test
  - `revenue-recognition` — Revenue cycle controls (order-to-cash)
  - `procure-to-pay` or `p2p` — Procurement and AP controls
  - `payroll` — Payroll processing and compensation controls
  - `financial-close` — Period-end close and reporting controls
  - `treasury` — Cash management and treasury controls
  - `fixed-assets` — Capital asset lifecycle controls

  - `inventory` — Inventory valuation and management controls
  - `itgc` — IT general controls (access, change management)
  - `entity-level` — Entity-level and monitoring controls
  - `journal-entries` — Journal entry processing controls
  - Or any specific control ID
- `period` — The testing period (e.g., `2024-Q4`, `2024`, `2024-H2`)

Examples:

```
/sox-testing revenue-recognition 2024-Q4
/sox-testing p2p 2024
/sox-testing itgc 2024-Q4
```

Workflow:

1. **Identify controls to test** — Present control matrix for the selected area with control descriptions, types, frequencies, and assertions.
2. **Determine sample size** — Calculate sample sizes based on control frequency and risk level.
3. **Generate sample selection** — Select samples using random, systematic, or targeted selection methodology.
4. **Create testing workpaper** — Generate a testing template with test objectives, procedures, expected evidence, and results documentation.
5. **Provide control templates** — Pre-built test step templates for common controls in the area.
6. **Document deficiency framework** — If exceptions are found, provide classification guidance (deficiency, significant deficiency, material weakness).

Output includes:

- Control matrix for the selected area
- Sample selection with methodology documentation
- Testing workpaper templates with pre-populated test steps
- Expected evidence descriptions
- Results documentation table
- Deficiency classification framework
- Remediation action framework

When to use this command:

- Annual SOX 404 testing cycle
- Interim control testing (for continuous monitoring programs)

- Preparing for external auditor walkthroughs
- Documenting control assessments for internal audit or management
- Generating workpapers for a new control or changed process

4. Skills Deep Dive

Skills are packages of domain knowledge that load automatically when relevant topics arise in conversation. Understanding what each skill contains helps you know when and how to leverage the plugin's expertise.

4.1 Journal Entry Preparation

Trigger phrases: "prepare journal entry", "book accrual", "depreciation entry", "prepaid amortization", "payroll entry", "revenue recognition"

What this skill contains:

Standard accrual types and their entries:

- **AP accruals**: When to accrue, how to calculate (PO receipts, contracts, estimates), standard entry format (debit expense, credit accrued liabilities), reversal requirements
- **Fixed asset depreciation**: Depreciation methods (straight-line, declining balance, units of production), when to start/stop depreciating, common useful lives by asset class
- **Prepaid amortization**: Building amortization schedules, common prepaid categories (insurance, software, rent), when to accelerate amortization
- **Payroll accruals**: Salary accrual for partial pay periods, bonus accrual methodology, benefits and tax accruals, PTO liability accrual
- **Revenue recognition**: ASC 606 five-step framework, identifying performance obligations, allocation of transaction price, timing of revenue recognition

Supporting documentation requirements: Every journal entry must include:

- Entry description/memo
- Calculation support
- Source documents
- Period identification
- Preparer identification
- Approval evidence
- Reversal indicator

Review and approval workflows:

- Typical approval matrix by entry type and amount
- Review checklist (debits = credits, correct period, valid accounts, etc.)
- Common errors to check for (unbalanced entries, wrong period, wrong sign, duplicates, missing reversals, stale accruals)

When this skill is most valuable:

- Month-end and quarter-end close
- Training new accounting staff on entry preparation standards
- Standardizing journal entry formats across the team
- Preparing for audits (ensuring entries have proper support and documentation)

4.2 Reconciliation

Trigger phrases: "reconcile account", "bank reconciliation", "subledger reconciliation", "intercompany reconciliation", "reconciling items"

What this skill contains:

Reconciliation types:

- **GL to subledger**: AR, AP, fixed assets, inventory, prepaid, accrued liabilities
- **Bank reconciliation:** Standard format (bank balance to GL balance), outstanding checks, deposits in transit, unrecorded items
- **Intercompany reconciliation**: Ensuring balances net to zero in consolidation, resolving FX differences, identifying timing issues

Reconciling item categorization:

- **Category 1: Timing differences** (will clear without action) — outstanding checks, deposits in transit, in-transit transactions
- **Category 2: Adjustments required** (need journal entry) — unrecorded bank charges, unrecorded interest, recording errors, missing entries
- **Category 3: Requires investigation** (root cause unknown) — unidentified differences, disputed items, aged outstanding items

Aging analysis:

- Age buckets (0-30, 31-60, 61-90, 90+ days)
- Actions by age bucket (monitor, investigate, escalate, escalate to management)
- Trending analysis (period-over-period growth in reconciling items)

Escalation thresholds: Sample thresholds by individual item amount, total reconciling items, item age, and trending. (Note: These should be customized to your organization's materiality levels.)

Best practices:

- Timeliness (complete within close calendar deadlines)
- Completeness (all balance sheet accounts on a defined frequency)
- Documentation (preparer, reviewer, date, clear explanations)
- Segregation of duties
- Follow-through on open items
- Root cause analysis for recurring issues
- Standardization and retention

When this skill is most valuable:

- Every month-end and quarter-end close
- Investigating account balance discrepancies
- Training staff on reconciliation methodology
- Audit preparation (ensuring reconciliations are complete and well-documented)

4.3 Financial Statements

Trigger phrases: "income statement", "balance sheet", "cash flow statement", "financial statement", "P&L", "GAAP presentation", "flux analysis"

What this skill contains:

Standard formats:

- **Income statement**: Revenue, cost of revenue, gross profit, operating expenses (by function), operating income, other income/expense, income before taxes, tax expense, net income. Includes GAAP presentation requirements (ASC 220 / IAS 1).
- **Balance sheet**: Current assets, non-current assets, current liabilities, non-current liabilities, stockholders' equity. Includes GAAP presentation requirements (ASC 210 / IAS 1).
- **Cash flow statement**: Operating activities (indirect method), investing activities, financing activities. Includes GAAP presentation requirements (ASC 230 / IAS 7).

GAAP presentation requirements:

- Expense classification by function vs. nature
- Current vs. non-current classification

- Required disclosures (depreciation, stock-based compensation, leases, etc.)
- Discontinued operations, extraordinary items (prohibited)
- Cash equivalents definition
- Non-cash investing and financing activities

Common adjustments and reclassifications:

- Period-end adjustments (accruals, deferrals, depreciation, bad debt provision, inventory write-downs, FX revaluation, tax provision)
- Reclassifications (current/non-current, contra account netting, intercompany elimination, discontinued operations)

Flux analysis methodology:

- Variance calculation (dollar variance, percentage variance, basis point change for margins)
- Materiality thresholds (fixed dollar, percentage, combined, scaled by line item size)
- Variance decomposition (volume/quantity, rate/price, mix, new/discontinued items, one-time items, timing, currency)
- Investigation and narrative (quantify, identify favorable/unfavorable, decompose, explain business reason, assess trend)

When this skill is most valuable:

- Preparing monthly, quarterly, or annual financial statements
- Ensuring GAAP compliance in presentation
- Training finance staff on statement formats and requirements
- Preparing for audit or investor reporting
- Understanding what adjustments and reclassifications are standard vs. unusual

4.4 Variance Analysis

Trigger phrases: "variance analysis", "flux analysis", "budget variance", "explain variance", "waterfall analysis", "driver decomposition"

What this skill contains:

Variance decomposition techniques:

- **Price/volume decomposition**: Total variance = (Actual Volume - Budget Volume) × Budget Price + (Actual Price - Budget Price) × Actual Volume. Used for revenue, COGS, and any metric that is Price × Volume.

- **Rate/mix decomposition**: Separating the effect of blended rate changes from shifts in product/segment mix. Used for gross margin analysis.
- **Headcount/compensation decomposition**: Breaking down payroll variances into headcount variance, rate variance, mix variance, timing variance, and attrition impact.
- **Spend category decomposition**: For operating expenses — headcount-driven costs, volume-driven costs, discretionary spend, contractual/fixed costs, one-time/non-recurring, timing/phasing.

Materiality thresholds and investigation triggers:

- Setting thresholds based on financial statement materiality, line item size, volatility, and management attention
- Recommended threshold framework (by comparison type: actual vs budget, actual vs prior period, actual vs forecast, sequential)
- Investigation priority (largest dollar variance, largest percentage variance, unexpected direction, new variance, cumulative/trending)

Narrative generation for variance explanations:

- Structure: [Line item], [Favorable/Unfavorable], [$amount], [%], [Driver description], [Outlook], [Action]
- Narrative quality checklist (specific, quantified, causal, forward-looking, actionable, concise)
- Common anti-patterns to avoid (circular explanations, vague statements, no quantification)

Waterfall chart methodology:

- Data structure: starting value, drivers with signed amounts, ending value
- Text-based waterfall format (for when charts are not available)
- Bridge reconciliation table (showing each driver's contribution as % of variance)
- Best practices (5-8 drivers max, ordered by magnitude, verify reconciliation, color-code)

Budget vs actual vs forecast comparisons:

- When to use each comparison (actual vs budget for annual performance, actual vs forecast for operational management)
- Forecast accuracy analysis (MAPE calculation, tracking forecast accuracy over time)
- Variance trending (consistent favorable = sandbagging?, consistent unfavorable = execution issue?, growing unfavorable = deteriorating performance)

When this skill is most valuable:

- Explaining material variances to management, board, or investors
- Preparing variance commentary for earnings releases
- Root cause analysis for budget misses or unexpected results

- Improving forecasting and planning accuracy by analyzing past variances
- Training FP&A or accounting staff on variance decomposition techniques

4.5 Close Management

Trigger phrases: "month-end close", "close calendar", "close tasks", "close dependencies", "close status", "accelerate the close"

What this skill contains:

Month-end close checklist:

- **Pre-close (T-2 to T-0):** Send calendar reminders, confirm cut-off procedures, verify systems, preliminary bank recs, review open POs, confirm payroll schedule
- **Close Day 1 (T+1):** Sub-ledger processing, AP accruals, payroll entries, cash entries, intercompany transactions, bank reconciliation, depreciation, prepaid amortization
- **Close Day 2 (T+2):** Revenue recognition, remaining accruals, AR/AP reconciliations, inventory adjustments, FX revaluation, begin balance sheet recs
- **Close Day 3 (T+3):** Complete balance sheet recs, adjusting entries, intercompany reconciliation, preliminary trial balance and P&L, preliminary flux analysis
- **Close Day 4 (T+4):** Tax provision, equity roll-forward, soft close, draft financial statements, detailed flux analysis, management review
- **Close Day 5 (T+5):** Final adjustments, hard close, period lock, reporting package distribution, forecast updates, close retrospective

Task sequencing and dependencies:

- Five-level dependency map (Level 1 = no dependencies, Level 5 = depends on everything prior)
- Critical path identification (the minimum close duration based on dependent tasks)
- How to shorten the close (automate Level 1 entries, pre-reconcile during the month, parallel-process independent tasks, set hard deadlines)

Status tracking and reporting:

- Close status dashboard (task, owner, deadline, status, blocker, notes)
- Status definitions (not started, in progress, complete, blocked, at risk)
- Daily close status meeting format (review status, identify blockers, reassign/escalate, update timeline)
- Close metrics to track over time (close duration, # of adjusting entries after soft close, # of late tasks, # of reconciliation exceptions, # of restatements/corrections)

Common close activities by day (organized by typical 5-day and accelerated 3-day close calendars):

Close process improvement:

- Common bottlenecks and solutions (late AP accruals → continuous accrual estimation; manual JEs → automate recurring entries; slow reconciliations → continuous reconciliation; intercompany delays → stricter deadlines and automation; management review changes → improve preliminary review)
- Close retrospective questions (What went well? What took longer than expected? What blockers occurred? Any surprises in results? What can we automate?)

When this skill is most valuable:

- Planning and executing every month-end and quarter-end close
- Training new accounting managers on close coordination
- Accelerating the close timeline (moving from 7-day to 5-day to 3-day close)
- Identifying close process improvements
- Managing close status and escalating blockers in real-time

4.6 Audit Support

Trigger phrases: "SOX testing", "control testing", "sample selection", "audit workpaper", "deficiency", "significant deficiency", "material weakness"

What this skill contains:

SOX 404 control testing methodology:

- Overview of the SOX 404 process (scoping, risk assessment, control identification, testing, evaluation, reporting)
- Scoping significant accounts (quantitative and qualitative factors)
- Relevant assertions by account type (revenue, AR, inventory, fixed assets, AP, accrued liabilities, equity, financial close)
- Design effectiveness vs operating effectiveness (walkthroughs vs testing)

Sample selection approaches:

- **Random selection**: When to use (transaction-level controls, large populations), method (random number generator), advantages/disadvantages
- **Targeted (judgmental) selection**: When to use (supplement to random, risk-based testing), method (select high-risk items), advantages/disadvantages

- **Haphazard selection**: When to use (no sequential population listing), method (select without pattern), advantages/disadvantages
- **Systematic selection**: When to use (sequential population), method (every Nth item), advantages/disadvantages
- **Sample size guidance**: Table showing recommended sample sizes by control frequency (annual, quarterly, monthly, weekly, daily, per-transaction) and risk level (low, moderate, high)

Testing documentation standards:

- Workpaper requirements (control identification, test design, test execution, conclusion, sign-off)
- Evidence standards (sufficient vs insufficient evidence)
- Working paper organization (by control area, with control matrix, walkthrough docs, test workpapers, supporting evidence, summary and conclusions)

Control deficiency classification:

- **Deficiency**: Control does not allow prevention or detection of misstatements on a timely basis. Evaluation factors: likelihood, magnitude, compensating controls.
- **Significant deficiency**: Less severe than a material weakness but important enough to merit attention by governance.
- **Material weakness**: Reasonable possibility that a material misstatement will not be prevented or detected.
- Deficiency aggregation (individual deficiencies may be significant in combination)
- Remediation framework (root cause analysis, remediation plan, timeline, owner, validation)

Common control types:

- **IT general controls (ITGCs)**: Access controls (provisioning, de-provisioning, privileged access, periodic reviews), change management (approval, testing, separation of dev/prod), IT operations (batch monitoring, backup/recovery, incident management)
- **Manual controls**: Examples (management review, journal entry approval, three-way match, account reconciliation), key attributes to test (right person, timely, evidence of review, sufficient information, exceptions addressed)
- **Automated controls**: Examples (system-enforced approvals, three-way match automation, duplicate payment detection, credit limit enforcement), testing approach (test design, test configuration, verify change management ITGCs)
- **IT-dependent manual controls**: Examples (management review of exception report, supervisor review of aging report), testing approach (test the manual control AND test completeness/accuracy of the underlying report/data)

- **Entity-level controls**: Examples (tone at the top, risk assessment, audit committee oversight, fraud risk assessment, monitoring controls, financial reporting competence), significance (can mitigate but not replace process-level controls; ineffective entity-level controls are strong indicators of material weakness)

When this skill is most valuable:

- Annual SOX 404 testing cycle
- Preparing for external auditor walkthroughs and testing
- Training internal audit or accounting staff on control testing
- Evaluating control deficiencies and determining severity
- Designing remediation plans for identified deficiencies
- Understanding what makes a control testable and what evidence is required

---

5. Integration and MCP Connectors

The finance plugin achieves its full potential when connected to your financial data sources and communication tools via MCP (Model Context Protocol) servers. While the plugin works without integrations (you can paste data or upload files), MCP connections enable automatic data retrieval, streamlined workflows, and real-time collaboration.

Pre-configured MCP Servers

The plugin's `.mcp.json` file includes connections to the following services:

SERVER	TYPE	URL	WHAT IT PROVIDES
Snowflake	HTTP	(placeholder)	Data warehouse queries for financial data, variance analysis, historical comparisons
Databricks	HTTP	(placeholder)	Data warehouse queries for financial data, analytics
BigQuery	HTTP	`https://bigquery.googleapis.com/mcp`	Data warehouse queries for financial data
Slack	HTTP	`https://mcp.slack.com/mcp`	Team communication for close status updates, approvals, questions

SERVER	TYPE	URL	WHAT IT PROVIDES
Microsoft 365	HTTP	`https://microsoft365.mcp.claude.com/mcp`	Email (send reports, request approvals), Excel (generate workpapers)

Note: Snowflake and Databricks URLs are placeholders — the MCP URL is not yet configured in the plugin's `.mcp.json`. You will need to configure these connections manually if you use these platforms.

Critical Missing Connectors

ERP / Accounting System: There is currently no standard MCP server for major ERP and accounting systems (NetSuite, SAP, QuickBooks, Xero). This is a significant gap. Without an ERP connector, you must:

- Manually export trial balances, subledger data, and journal entries from your ERP
- Paste the data into the conversation or upload as files
- Manually post journal entries generated by the plugin back into your ERP

Analytics / BI Tools: Similarly, there are no pre-configured MCP servers for major BI platforms (Tableau, Looker, Power BI). For variance analysis that leverages BI dashboards, you must manually export data or screenshots.

How to Connect MCP Servers

For HTTP servers (like BigQuery, Slack, Microsoft 365):

1. The plugin's `.mcp.json` already includes the URL
2. When you first use a command that references the tool (e.g., posting to Slack), you will be prompted to authenticate
3. After authentication, the connection is maintained automatically

For stdio servers (local tools or custom integrations):

1. Edit the plugin's `.mcp.json` file to add your custom server
2. Provide the command to run, arguments, and any required environment variables
3. Example for a custom NetSuite integration:

```
{
  "mcpServers": {
    "netsuite-local": {
      "command": "node",
      "args": ["${CLAUDE_PLUGIN_ROOT}/custom-servers/netsuite-mcp.js"],
      "env": {
        "NETSUITE_ACCOUNT_ID": "${NETSUITE_ACCOUNT_ID}",
        "NETSUITE_API_KEY": "${NETSUITE_API_KEY}"
      }
    }
  }
}
```

For missing ERP/BI connectors:

- Check if your vendor provides an MCP server (this is an emerging standard, and more vendors are adding MCP support)
- If not, consider building a simple stdio MCP server that wraps your ERP's REST API
- As a workaround, use the data warehouse connector if your ERP data is replicated to your warehouse

Data Warehouse Integration Patterns

If you have an ERP-to-warehouse replication pipeline (common for companies using Fivetran, Stitch, or custom ETL), the data warehouse connector becomes your primary data source for the finance plugin.

Typical queries the plugin will run:

- Pull trial balance for a specific period
- Pull subledger detail for a specific account
- Pull historical financial data for variance analysis
- Pull operational metrics (headcount, units sold, ASP) for variance decomposition

Setting up your warehouse for finance plugin use:

1. Ensure your ERP data is replicated to the warehouse on a schedule that supports your close timeline (ideally daily or more frequently)
2. Create views or tables that present financial data in a query-friendly format:
    - Trial balance view (account code, account name, period, debit/credit balance)
    - Journal entry view (entry ID, date, period, account, debit, credit, description)
    - Subledger views (AR aging, AP aging, fixed asset register, prepaid schedule, accrual schedule)

3. Grant the MCP server read-only access to these views
4. Document the schema so Claude can construct correct queries (alternatively, provide sample queries in your project's `CLAUDE.md`)

Slack Integration for Close Management

Connecting Slack enables real-time close status updates and team communication:

Typical workflows:

- Post daily close status summaries to a dedicated close channel (e.g., `#month-end-close`)
- Send @mentions to task owners when tasks are blocked or overdue
- Request approvals for journal entries or adjusting entries from the controller or finance director
- Announce when soft close or hard close is achieved

Setup:

1. Authenticate the Slack MCP server (prompted on first use)
2. Create a dedicated close channel in Slack (e.g., `#month-end-close`, `#q4-close`)
3. In your project's `CLAUDE.md` or in close management commands, specify the channel name for status updates

Email Integration for Report Distribution

Connecting Microsoft 365 email enables automated report distribution:

Typical workflows:

- Email the monthly financial reporting package to management and board
- Send variance analysis reports to department heads for their review
- Request supporting documentation from business unit controllers
- Distribute reconciliation workpapers to reconciliation preparers

Setup:

1. Authenticate the Microsoft 365 MCP server (prompted on first use)
2. The plugin can then compose and send emails with attachments (financial statements, variance reports, workpapers)

Excel / Spreadsheet Integration

The Microsoft 365 MCP server also provides access to Excel for workpaper generation:

Typical workflows:

- Generate reconciliation workpapers in Excel format
- Create variance analysis tables with formulas and formatting
- Build depreciation schedules or prepaid amortization schedules

Note: If your organization uses Google Sheets instead of Excel, you will need to configure a Google Workspace MCP server (not pre-configured in the plugin).

Security and Access Control

Best practices for MCP connections:

- Use environment variables for credentials (e.g., `${NETSUITE_API_KEY}`) — never hard-code credentials in `.mcp.json`
- Grant read-only access whenever possible (especially for data warehouse queries)
- Use separate service accounts for MCP server access (not personal accounts)
- Monitor MCP server usage and audit logs to ensure appropriate use
- Follow your organization's data access and security policies

6. Workflows and Use Cases

This section walks through real-world workflows that combine multiple commands and skills to accomplish complete finance and accounting tasks.

Workflow 1: Complete Month-End Close (5-Day Calendar)

Scenario: It is the end of December 2024. You are the accounting manager coordinating the close for your team of four accountants. Your target is T+5 hard close (close by January 7, 2025).

Day T+1 (January 2, 2025) — Journal Entries and Initial Reconciliations

1. **Generate AP accrual:**

   ```
   /journal-entry ap-accrual 2024-12
   ```

   Review the accrual calculation, verify supporting detail, post to the GL.

2. **Generate depreciation entry:**

```
/journal-entry fixed-assets 2024-12
```

Verify depreciation by asset class, post to the GL.

3. **Generate prepaid amortization entry:**

   ```
   /journal-entry prepaid 2024-12
   ```

   Verify amortization schedule, post to the GL.

4. **Generate payroll accrual** (if pay period straddles month-end):

   ```
   /journal-entry payroll 2024-12
   ```

   Verify days worked vs. pay period, post to the GL.

5. **Reconcile cash:**

   ```
   /reconciliation cash 2024-12-31
   ```

   Review outstanding checks and deposits in transit, verify no large or aged items, document reconciliation.

Day T+2 (January 3, 2025) — Revenue, Remaining Accruals, Subledger Reconciliations

6. **Generate revenue recognition entry:**

   ```
   /journal-entry revenue 2024-12
   ```

   Review performance obligations, verify delivery/completion, post to the GL.

7. **Reconcile AR subledger:**

   ```
   /reconciliation ar 2024-12
   ```

   Verify GL control account matches AR aging, investigate any differences, document reconciliation.

8. **Reconcile AP subledger:**

```
/reconciliation ap 2024-12
```

Verify GL control account matches AP aging, investigate any differences, document reconciliation.

Day T+3 (January 6, 2025) — Balance Sheet Reconciliations, Preliminary P&L

9. **Reconcile all balance sheet accounts** (use `/reconciliation` for each material account or assign to team members).

10. **Generate preliminary income statement:**

    ```
    /income-statement monthly 2024-12
    ```

    Review the P&L, flag material variances, identify areas for deeper analysis.

11. **Investigate material variances:** For each flagged variance from the income statement:

    ```
    /variance-analysis revenue 2024-12 vs 2024-11
    /variance-analysis opex 2024-12 vs budget
    ```

    Document variance drivers and prepare explanations.

Day T+4 (January 7, 2025) — Tax Provision, Soft Close, Management Review

12. **Post tax provision** (prepare manually or using your tax software, then post to GL).

13. **Generate final income statement for management review:**

    ```
    /income-statement monthly 2024-12
    ```

14. **Prepare variance commentary** (using the outputs from `/variance-analysis` on Day T+3).

15. **Management review:** Present the December P&L, variance explanations, and any required adjusting entries.

Day T+5 (January 8, 2025) — Hard Close, Reporting Package

16. **Post final adjusting entries** (from management review).

17. **Lock the period** in your ERP/GL system.

18. **Generate final financial reporting package:**

```
/income-statement monthly 2024-12
```

Combine with balance sheet, cash flow statement, and variance commentary.

19. **Distribute reporting package** via email or Slack (if Microsoft 365 or Slack MCP is connected).
20. **Update forecast/budget** based on actual results.
21. **Close retrospective:** Ask the team what went well, what took longer than expected, and what can be improved for next month.

Workflow 2: Quarterly Variance Analysis for Board Reporting

Scenario: It is early January 2025. You are the controller preparing Q4 2024 results for the board meeting on January 15. The board wants to understand why Q4 revenue was below the forecast and why operating expenses were higher than expected.

Step 1: Generate Q4 Income Statement

```
/income-statement quarterly 2024-Q4
```

Review the output. Note that revenue is 8% below forecast and operating expenses are 12% above forecast.

Step 2: Analyze Revenue Variance

```
/variance-analysis revenue 2024-Q4 vs forecast
```

The plugin decomposes the revenue variance into drivers:

- Volume effect: -$500K (lower customer count)
- Price effect: +$150K (higher ASP)
- Mix effect: -$200K (shift toward lower-margin products)
- Deal slippage: -$300K (large enterprise deals delayed to Q1)
- Net variance: -$850K

Document the narrative:

> **Revenue shortfall of $850K (8% below forecast)**_: The primary driver was lower-than-expected customer acquisition ($500K), partially offset by higher average selling prices (+$150K). Additionally, two large enterprise deals totaling $300K slipped from Q4 to Q1 due to extended procurement cycles at the customer. The product mix shift toward lower-priced SKUs contributed an additional $200K unfavorable variance. Outlook: The slipped enterprise deals are expected to close in Q1, and customer acquisition is projected to improve with the new marketing campaign launching in January._

Step 3: Analyze Operating Expense Variance

```
/variance-analysis opex 2024-Q4 vs forecast
```

The plugin decomposes the opex variance into drivers:

- Headcount-driven: +$200K (hired 3 engineers ahead of plan)
- Volume-driven: +$150K (higher cloud hosting due to usage spike)
- Discretionary spend: +$300K (unplanned conference attendance and legal fees)
- One-time items: +$100K (severance for departing VP)
- Timing: -$50K (some January spend pulled into December)
- Net variance: +$700K

Document the narrative:

> **Operating expense overrun of $700K (12% above forecast)**_: The largest driver was unplanned discretionary spend ($300K), primarily legal fees for a contract dispute and unbudgeted conference attendance. Headcount additions ahead of plan (3 engineers hired in Q4 vs. Q1 plan) contributed $200K, and cloud hosting costs increased $150K due to higher-than-expected usage. A one-time severance charge ($100K) for a departing VP also contributed. Outlook: Discretionary spend is expected to normalize in Q1, and headcount is now aligned with the updated hiring plan. Cloud hosting costs remain elevated but are expected to stabilize._

Step 4: Prepare Board Presentation

Combine the income statement, variance waterfall charts (if using a charting tool), and narrative explanations into a concise board presentation. Include:

- Q4 actual vs. forecast summary (revenue, gross profit, operating expenses, net income)
- Revenue variance waterfall with narrative

- Opex variance waterfall with narrative
- Forward-looking commentary (Q1 outlook, full-year implications)

Workflow 3: SOX 404 Testing for Revenue Recognition Controls

Scenario: It is November 2024, and you are the internal auditor responsible for testing revenue recognition controls for the full year 2024. You need to generate sample selections, create testing workpapers, and document your test results.

Step 1: Generate Control Matrix and Sample Selection

```
/sox-testing revenue-recognition 2024
```

The plugin outputs:

- Control matrix for the revenue cycle (order approval, delivery verification, pricing accuracy, credit memo approval, revenue cut-off)
- Recommended sample sizes (25 samples for the order approval control, which operates per-transaction with a population of ~2,500 sales orders)
- Sample selection using random selection methodology (25 randomly selected sales orders from the population of 2,500)

Step 2: Generate Testing Workpaper

For each control, the plugin provides a pre-populated testing workpaper:

```
SOX CONTROL TESTING WORKPAPER
Control #: REV-001
Control Description: Sales orders require approval by the sales manager before invoici
Control Owner: Sales Manager
Control Type: Manual
Frequency: Per transaction
Key Control: Yes
Relevant Assertion: Occurrence, Accuracy

TEST OBJECTIVE:
To determine whether sales orders were approved by the sales manager before invoicing

TEST PROCEDURES:
1. Obtain the sales order for each sample item
2. Verify that the sales order includes a sales manager approval (signature or electro
3. Verify that the approval date is prior to the invoice date
4. Verify that the approver has the authority to approve sales orders (check approval
5. Verify that the order amount is within the approver's authority limit

EXPECTED EVIDENCE:
- Signed sales order form or electronic approval record in the CRM
- Approval date stamp
- Approver name and title

TEST RESULTS:
[Table for documenting pass/fail for each sample, with columns for Sample #, Ref, Proc
```

Step 3: Execute Testing

For each sample:

1. Pull the sales order from the CRM or order management system
2. Verify the sales manager approval signature/timestamp
3. Verify the approval date is before the invoice date
4. Check the approval matrix to confirm the approver has authority
5. Document pass/fail in the testing workpaper

Step 4: Document Exceptions and Conclusion

If exceptions are found (e.g., 2 out of 25 samples lack sales manager approval):

- Document the exception description (which samples, what was missing)
- Perform root cause analysis (Was the control not performed? Was there a system issue? Was training inadequate?)
- Evaluate whether compensating controls exist

- Classify the deficiency (deficiency, significant deficiency, or material weakness)
- Prepare a remediation plan

The plugin provides the deficiency classification framework:

- 2 out of 25 samples = 8% exception rate
- Given the key control designation and the lack of a compensating control, this is likely a **significant deficiency** (less severe than a material weakness but important enough to merit attention)
- Remediation: Implement a system-enforced approval workflow that prevents invoicing without sales manager approval; retrain sales operations team on approval requirements; re-test the control in Q1 2025 to verify effectiveness.

Step 5: Prepare Summary for Management

Compile all testing workpapers, sample selections, and deficiency documentation into a summary report for the controller and CFO. Include:

- Controls tested and sample sizes
- Test results (pass/fail by control)
- Deficiencies identified with classification and remediation plans
- Overall conclusion on the effectiveness of revenue recognition controls

---

Workflow 4: Accelerating the Close (Moving from 5-Day to 3-Day Close)

Scenario: Your organization currently closes in 5 business days (T+5). The CFO has set a goal to reduce this to 3 business days (T+3) by the end of Q1 2025. You are the accounting manager tasked with identifying opportunities to accelerate the close.

Step 1: Analyze Current Close Bottlenecks

Ask the close-management skill for common bottlenecks:

> *"What are the most common bottlenecks in the month-end close, and how can I address them?"*

The plugin identifies:

- Late AP accruals (waiting for department spend confirmation)
- Manual recurring journal entries (depreciation, prepaid amortization, standard accruals)
- Slow reconciliations (starting from scratch each month)

- Intercompany delays (waiting for counterparty confirmation)
- Management review changes (large adjustments found during review)

Step 2: Implement Automation for Recurring Entries

For depreciation and prepaid amortization:

1. Automate these entries in your ERP system (most ERPs support scheduled recurring entries)
2. Set up the depreciation run to execute automatically on the first business day after month-end
3. Set up prepaid amortization to post automatically based on the prepaid schedule

For standard accruals (e.g., recurring vendor accruals):

1. Create templates for recurring accruals
2. Pre-populate amounts based on contracts or historical run rates
3. Only update if amounts change materially

Step 3: Implement Continuous Reconciliation

For cash, AR, and AP reconciliations:

1. Reconcile weekly instead of monthly (this reduces the volume of items to investigate at month-end)
2. Use automated reconciliation matching tools (if your ERP or treasury system supports it)
3. Maintain a rolling list of outstanding reconciling items (update weekly, so at month-end you only need to add items from the last few days)

Step 4: Tighten Intercompany Cut-Off and Automation

For intercompany transactions:

1. Set a hard cut-off deadline for intercompany transactions (e.g., all intercompany transactions must be posted by T+1 noon)
2. Automate intercompany elimination entries (most consolidation systems support this)
3. Implement daily intercompany reconciliation (so any discrepancies are caught and resolved before month-end)

Step 5: Improve Preliminary Review Process

To reduce adjustments found during management review:

1. Run a preliminary P&L on T+3 afternoon (before final review)
2. Perform preliminary flux analysis and identify any unusual variances
3. Investigate and resolve variances before presenting to management (this reduces the likelihood of adjustments during review)

Step 6: Update Close Calendar for 3-Day Close

Redesign the close calendar with the automation and process improvements:

DAY	KEY ACTIVITIES
T+1	All automated JEs post, all manual accruals posted by noon, all subledger reconciliations complete, bank reconciliation complete, intercompany reconciliation complete by end of day, preliminary trial balance
T+2	All balance sheet reconciliations complete, adjusting entries from reconciliations posted, tax provision posted, consolidation and eliminations complete, draft financial statements, detailed flux analysis, management review by end of day
T+3	Final adjustments from management review, hard close by noon, period lock, reporting package distribution, forecast update, close retrospective

Step 7: Pilot the 3-Day Close

Run the first 3-day close in January 2025:

1. Communicate the new close calendar and deadlines to the team
2. Monitor close status in real-time (use `/close-management` skill for task tracking)
3. Identify any tasks that miss the new deadlines and investigate why
4. Conduct a close retrospective to identify further improvements

Step 8: Iterate and Refine

After 2-3 months of the 3-day close:

1. Track close metrics (close duration, # of late tasks, # of adjustments after soft close)
2. Identify any remaining bottlenecks
3. Continue to refine and automate
4. Celebrate the team's achievement

7. Customization and Extension

The finance plugin is designed to be customized for your organization's specific needs. This section walks through common customization scenarios and the best approach for each.

Customization Approach Decision Tree

Question 1: Is this customization for your entire organization, or just a specific project/team?

- **Entire organization** → Fork the plugin (Approach B)
- **Specific project/team** → Use project-level skills and CLAUDE.md rules (Approach C)

Question 2: Does this customization change the plugin's core behavior, or does it add new capabilities?

- **Changes core behavior** (e.g., different entry formats, different reconciliation methodology) → Fork the plugin (Approach B)
- **Adds new capabilities** (e.g., new entry types, industry-specific rules) → Fork the plugin or use project-level skills (Approach B or C)

Question 3: Do you need to receive future plugin updates without merge conflicts?

- **Yes** → Use project-level skills and CLAUDE.md rules (Approach C)
- **No, I can manage updates manually** → Fork the plugin (Approach B)

Customization Scenario 1: Adding Industry-Specific Entry Types (SaaS Revenue Recognition)

Need: Your company is a SaaS business with subscription revenue. You need journal entry templates for:

- Deferred revenue adjustments (monthly recognition of annual contracts)
- Unbilled revenue / accounts receivable for usage-based billing
- Contract liability and contract asset tracking

Approach: Fork the plugin and add a new entry type to the `journal-entry` command.

Steps:

1. **Copy the plugin directory:**

```
finance/  →  finance-saas/
```

2. **Update plugin.json:**

```
{
  "name": "finance-saas",
  "version": "1.0.0",
  "description": "Finance plugin adapted for SaaS and subscription businesses."
}
```

3. **Modify** `commands/journal-entry.md` to add the new entry type:

   In the **Arguments** section, add:

```
- `type` — The journal entry type. One of:
  - `ap-accrual` — Accounts payable accruals
  - `fixed-assets` — Depreciation and amortization
  - `prepaid` — Amortization of prepaid expenses
  - `payroll` — Payroll accruals
  - `revenue` — Revenue recognition entries
  - `deferred-revenue` — Monthly deferred revenue recognition for subscription cont
  - `unbilled-revenue` — Unbilled revenue for usage-based billing
```

   In the **Workflow** section, add under "Calculate the Entry":

```
**Deferred Revenue (SaaS):**
- Pull the deferred revenue schedule (contract-level detail)
- Calculate the period revenue to recognize for each contract (contract value / con
- Debit: Deferred revenue
- Credit: Revenue (by product line or revenue stream)

**Unbilled Revenue (Usage-Based):**
- Pull usage data for the period (from billing system or product analytics)
- Calculate revenue for usage not yet invoiced (usage × rate)
- Debit: Unbilled revenue (accounts receivable)
- Credit: Revenue
```

4. **Add a new skill** `skills/saas-revenue/SKILL.md`:

```
---
name: saas-revenue
description: >
  SaaS and subscription revenue recognition methodology. Use when
  booking deferred revenue, recognizing subscription revenue, handling
  usage-based billing, or preparing contract asset/liability schedules
  for SaaS businesses.
---

# SaaS Revenue Recognition
```

```
## ASC 606 Application for SaaS

SaaS revenue recognition follows ASC 606 with specific considerations
for subscription models:

### Performance Obligations
- **Software access (SaaS subscription)**: Recognized ratably over the subscription
- **Implementation / onboarding services**: Recognized as services are delivered (i
- **Training**: Recognized when training is delivered
- **Premium support**: Recognized ratably over the support term

### Contract Modifications
- **Upgrades/downgrades mid-term**: Prospective treatment (adjust remaining revenue
- **Early renewals**: Treat as new contract or modification depending on pricing
- **Cancellations**: Cease revenue recognition; evaluate refund obligations

## Deferred Revenue Schedule

Maintain a contract-level deferred revenue schedule with:
- Contract ID and customer name
- Contract start and end dates
- Total contract value
- Monthly recognition amount
- Cumulative revenue recognized to date
- Remaining deferred revenue balance

### Monthly Entry Process
1. Run the deferred revenue schedule to calculate the period's recognition
2. Debit: Deferred revenue (by contract or in total)
3. Credit: Revenue (by product line)
4. Verify the remaining deferred revenue balance ties to the GL

## Unbilled Revenue (Usage-Based Models)

For usage-based billing (e.g., API calls, compute hours, data storage):
- Pull usage data as of period end from the product
- Calculate revenue for usage not yet invoiced (usage × rate)
- Book unbilled revenue (AR) entry
- When invoice is generated (next period), reverse unbilled revenue and record stan

## Contract Assets and Contract Liabilities

- **Contract asset**: Revenue recognized exceeds amounts invoiced (common when upfr
- **Contract liability**: Amounts invoiced exceed revenue recognized (deferred reve

Balance sheet presentation:
- Contract assets: Presented separately from AR (or disclosed in notes)
- Contract liabilities: Presented separately from deferred revenue (or combined wit
```

5. **Install the forked plugin:** Place `finance-saas/` in your plugins directory or package as `finance-saas.plugin`.
6. **Test the new entry type:**

   ```
   /journal-entry deferred-revenue 2024-12
   ```

Customization Scenario 2: Adding Company-Specific Approval Thresholds

Need: Your company has specific journal entry approval thresholds that differ from the generic examples in the plugin. You want all journal entry commands to apply your company's approval matrix automatically.

Approach: Use project-level `CLAUDE.md` rules (Approach C).

Steps:

1. **Document your approval matrix** in your project's `CLAUDE.md`:

```
## Acme Corp Finance Rules

### Journal Entry Approval Matrix

All journal entries must follow this approval chain based on amount:

| Entry Amount | Approver | Turnaround SLA |
├──────────────┼──────────┼────────────────┤
| Under $10K | Senior Accountant | 1 business day |
| $10K - $50K | Accounting Manager | 2 business days |
| $50K - $250K | Controller | 3 business days |
| Over $250K | CFO | 5 business days |

**Special rules**:
- Any out-of-period adjustments (posting to a prior period): Require Controller app
- Any top-side consolidation entries: Require Controller approval
- Any related-party transaction entries: Require CFO approval

### Account Code Standards

- Use 4-digit account codes, not account names, in all journal entries
- Always include department code (2-digit) for expense entries
- Revenue entries must include revenue stream code (product, service, other)

### Close Calendar Deadlines

- AP accruals due: T+1 by 2pm
- All manual journal entries due: T+2 by 5pm
- Balance sheet reconciliations due: T+3 by 5pm
- Soft close: T+4 by noon
- Hard close: T+5 by 5pm

When preparing journal entries or coordinating close tasks, always apply these Acme
```

2. **Test the customization:**

```
/journal-entry ap-accrual 2024-12
```

The plugin should now include Acme Corp's approval requirements in the journal entry output:

```
Approval Required: Accounting Manager (entry amount $15,000 is in the $10K-$50K tie
Turnaround SLA: 2 business days
```

Customization Scenario 3: Integrating Company-Specific ERP Data

Need: Your company uses NetSuite as its ERP. You want the finance plugin to pull trial balance and subledger data directly from NetSuite instead of requiring manual data exports.

Approach: Create a custom stdio MCP server for NetSuite (if a standard NetSuite MCP server is not available).

Steps:

1. **Build a simple NetSuite MCP server** (requires Node.js and NetSuite API knowledge):

   Create `custom-servers/netsuite-mcp.js`:

```
// Simplified example - not production-ready
const { Server } = require("@modelcontextprotocol/sdk/server");
const axios = require("axios");

const server = new Server({
  name: "netsuite-mcp",
  version: "0.1.0"
});

server.tool("get_trial_balance", async ({ period }) => {
  // Call NetSuite REST API to pull trial balance for the period
  const response = await axios.get(`https://${process.env.NETSUITE_ACCOUNT_ID}.suit
    headers: {
      "Authorization": `Bearer ${process.env.NETSUITE_ACCESS_TOKEN}`,
      "Content-Type": "application/json"
    },
    params: { period }
  });

  return {
    content: [
      {
        type: "text",
        text: JSON.stringify(response.data, null, 2)
      }
    ]
  };
});

server.start();
```

2. **Add the MCP server to the plugin's `.mcp.json`:**

```
{
  "mcpServers": {
    "netsuite": {
      "command": "node",
      "args": ["${CLAUDE_PLUGIN_ROOT}/custom-servers/netsuite-mcp.js"],
      "env": {
        "NETSUITE_ACCOUNT_ID": "${NETSUITE_ACCOUNT_ID}",
        "NETSUITE_ACCESS_TOKEN": "${NETSUITE_ACCESS_TOKEN}"
      }
    }
  }
}
```

3. **Set environment variables** with your NetSuite credentials:

```
export NETSUITE_ACCOUNT_ID="123456"
export NETSUITE_ACCESS_TOKEN="your_access_token_here"
```

4. **Test the integration:**

```
/journal-entry ap-accrual 2024-12
```

The plugin should now pull the trial balance from NetSuite automatically (via the `netsuite` MCP server's `get_trial_balance` tool).

Customization Scenario 4: Adding Custom Materiality Thresholds for Variance Analysis

Need: Your company defines materiality as 1% of revenue (not a fixed dollar amount). You want variance analysis to use this dynamic threshold instead of the generic thresholds in the plugin.

Approach: Use project-level `CLAUDE.md` rules (Approach C).

Steps:

1. **Document your materiality policy** in your project's `CLAUDE.md`:

```
## Acme Corp Materiality Thresholds

### Financial Statement Materiality
- **Planning materiality**: 1% of total revenue (for annual financial statements)
- **Performance materiality**: 75% of planning materiality
- **Clearly trivial threshold**: 3% of planning materiality

### Variance Analysis Thresholds
For income statement line items, flag variances that meet either condition:
- Dollar variance exceeds 0.1% of total revenue (1/10th of planning materiality)
- Percentage variance exceeds 10%

For balance sheet line items, flag variances that meet either condition:
- Dollar variance exceeds 0.5% of total assets
- Percentage variance exceeds 15%

### Example Calculation (based on 2024 revenue of $50M):
- Planning materiality: $500K (1% of $50M)
- Income statement variance threshold: $50K (0.1% of $50M) OR 10%
- Balance sheet variance threshold: Assume total assets of $100M → $500K (0.5% of $1

When performing variance analysis, always apply Acme Corp's materiality thresholds.
```

2. **Test the customization:**

```
/income-statement monthly 2024-12
```

The plugin should now flag variances using your dynamic thresholds (e.g., if December revenue was $4.2M, the threshold would be $4,200 or 10%, whichever is met first).

Best Practices for Customization

1. **Document your customizations:** Maintain a `CUSTOMIZATIONS.md` file in your forked plugin or project that lists all changes from the base plugin.

2. **Version your forked plugins**: Use semantic versioning (e.g., `1.0.0`, `1.1.0`) and update the version in `plugin.json` whenever you make changes.

3. **Test thoroughly**: After customizing, test all affected commands and skills to ensure the changes work as expected.

4. **Keep customizations minimal**: Only customize what is truly unique to your organization. Rely on the base plugin for standard methodologies.

5. **Share customizations with your team**: If multiple people use the plugin, ensure everyone uses the same customized version (distribute the forked `.plugin` file).
6. **Monitor for base plugin updates**: If you fork the plugin, periodically check for updates to the base plugin and manually merge improvements.

AGENT_REPORT_START Plugin: finance Version: 1.0.0 Files analyzed: 17

- plugin.json
- .mcp.json
- CONNECTORS.md
- README.md
- 5 command files (journal-entry.md, reconciliation.md, income-statement.md, variance-analysis.md, sox-testing.md)
- 6 skill files (journal-entry-prep/SKILL.md, reconciliation/SKILL.md, financial-statements/SKILL.md, variance-analysis/SKILL.md, close-management/SKILL.md, audit-support/SKILL.md)
- Plugin development guide (00-plugin-development-guide.md)

Handbook word count: 10,500 (approximately) Start time: 1771068395 End time: 1771068950 Estimated token usage: 80,000 tokens AGENT_REPORT_END

RationalEyes.ai

contact@rationaleyes.ai

Intelligent Automation for Knowledge Work

Finance & Accounting Plugin Handbook — Version 1.0.0