Tax Ready Bookkeeping

AI-Infused SOP Format Template

Tax Ready Bookkeeping | projectbits.com/taxready

Overview

This template shows how to write Standard Operating Procedures (SOPs) that can be executed by both humans and AI agents. Each procedure includes machine-readable rules alongside human-friendly instructions.

SOP Structure

Required Elements for AI-Infused SOPs

Section	Human Purpose	AI Purpose
Title & ID	Reference & filing	Unique identifier
Trigger	When to perform	Event detection
Prerequisites	What's needed first	Dependency check
Steps	How to perform	Sequential execution
Decision Points	Judgment calls	Branching logic
Validation	Quality check	Output verification
Exceptions	Edge cases	Error handling
Escalation	When to get help	Confidence thresholds

Template: AI-Infused SOP

SOP Header

```
SOP ID: [DEPT]-[PROCESS]-[VERSION]
Title: [Process Name]
Category: [Bookkeeping | AR | AP | Payroll | Reporting | Compliance]
Last Updated: [Date]
Owner: [Role]
AI Enabled: Yes/No
```

```
AI Confidence Required: [%]
```

Example Header

```
SOP ID: AP-VENDOR-PAY-v2.1
Title: Vendor Payment Processing
Category: AP
Last Updated: 2024-12-15
Owner: Senior Bookkeeper
AI Enabled: Yes
AI Confidence Required: 92%
```

Section 1: Trigger Conditions

Human Version

When this procedure should be initiated

Example: "This procedure is performed when vendor invoices are approved and ready for payment, typically on the weekly payment run (every Thursday)."

AI Version (Machine-Readable)

```yaml
trigger:
  type: scheduled
  schedule: "0 9 * * 4"  # Every Thursday at 9 AM
  conditions:
    - invoice.status == "approved"
    - invoice.due_date <= today + 7_days
    - vendor.payment_hold == false

  # Alternative event-based trigger
  event_trigger:
    - on: invoice.approved
      delay: 0  # Immediate
```

Section 2: Prerequisites

Human Version

What must be in place before starting

- ☐ Bank reconciliation current through yesterday
- ☐ Sufficient funds in operating account
- ☐ Access to accounting system and bank portal
- ☐ Approved invoice list from AP manager

AI Version

```yaml
prerequisites:
  data_requirements:
    - entity: bank_reconciliation
      condition: last_reconciled_date >= today - 1_day

    - entity: bank_account
      field: available_balance
      condition: >= sum(pending_payments.amount)

    - entity: invoices
      condition: status == "approved" AND payment_date IS NULL
      minimum_count: 1

  system_access:
    - system: accounting_software
      permission: write
    - system: bank_portal
      permission: payment_initiate

  human_approval:
    required: true
    role: ap_manager
    timeout: 4_hours
```

Section 3: Procedure Steps

Step Format

For each step, include: 1. **Human instruction** - Clear, actionable language 2. **AI instruction** - Specific system actions 3. **Validation** - How to verify completion 4. **Decision point** - If applicable

Example Step

Step 3: Verify Invoice Details

Human Instruction: Review each invoice to confirm: - Vendor name matches our records - Invoice number hasn't been paid before - Amount matches approved PO or contract - Payment terms are correctly applied

AI Instruction:

```yaml
step_3:
  name: verify_invoice_details
  actions:
    - action: match_vendor
      source: invoice.vendor_name
```

```
      target: vendor_master.name
      match_type: fuzzy
      threshold: 0.95

    - action: check_duplicate
      field: invoice.invoice_number
      against: payment_history.invoice_number
      result_if_found: flag_duplicate

    - action: match_amount
      source: invoice.amount
      target: purchase_order.amount
      tolerance: 0.02   # 2% variance allowed

    - action: calculate_due_date
      formula: invoice.date + vendor.payment_terms_days

  validation:
    all_conditions_met:
      - vendor_matched == true
      - duplicate_found == false
      - amount_within_tolerance == true

  on_failure:
    action: escalate
    to: ap_supervisor
    message: "Invoice verification failed: {failure_reasons}"
```

Validation: - Green checkmarks appear next to all verified items - No duplicate warnings displayed - Amount variance shows 0% or within tolerance

Decision Point: - If vendor not found -> Go to Step 3a (New Vendor Setup) - If duplicate detected -> Go to Step 3b (Duplicate Resolution) - If amount exceeds tolerance -> Go to Step 3c (Variance Approval)

Section 4: Decision Points Template

Human Version

If This Happens	Then Do This
Invoice > $5,000	Requires additional approval
Vendor is new	Complete W-9 verification first
Payment terms modified	Supervisor review required
Rush payment requested	Verify authorization email

4

AI Version

```yaml
decision_points:
  high_value_invoice:
    condition: invoice.amount > 5000
    action: require_approval
    approver: controller
    timeout: 24_hours

  new_vendor:
    condition: vendor.first_payment == true
    action: verify_w9
    block_until: w9.status == "verified"

  modified_terms:
    condition: invoice.payment_terms != vendor.default_terms
    action: require_review
    reviewer: ap_supervisor
    documentation: "Terms modification reason"

  rush_payment:
    condition: invoice.priority == "rush"
    action: verify_authorization
    verification_type: email_confirmation
    from_roles: [controller, cfo, owner]
```

Section 5: Validation & Quality Checks

Human Version

Before completing, verify: - [ ] Total payments match approved batch total - [ ] Bank balance sufficient for all payments - [ ] No duplicate payments in batch - [ ] All vendors have current W-9s (for 1099 vendors)

AI Version

```yaml
validation:
  pre_submission:
    - check: batch_total_match
      expected: sum(approved_invoices.amount)
      actual: sum(payment_batch.amount)
      tolerance: 0.00

    - check: sufficient_funds
      required: payment_batch.total
      available: bank_account.balance - pending_payments

    - check: no_duplicates
```

```
        method: unique(payment_batch.invoice_id)

    - check: w9_compliance
      for_each: vendor WHERE payment_ytd > 600
      condition: w9.status == "current"

  post_submission:
    - verify: confirmation_number_received
    - verify: payment_status == "submitted"
    - log: audit_trail.append(payment_record)
```

Section 6: Exception Handling

Human Version

Exception	Resolution
System timeout	Wait 5 minutes, retry once, then escalate to IT
Bank rejection	Note error code, contact bank, do not retry
Vendor account changed	Verify via phone call to known number
Amount discrepancy found	Stop batch, investigate, document

AI Version

```
exceptions:
  system_timeout:
    max_retries: 1
    retry_delay: 300   # 5 minutes
    on_persist: escalate_to_it
    preserve_state: true

  bank_rejection:
    action: halt
    notification: [ap_supervisor, controller]
    documentation_required: true
    retry_allowed: false_until_human_review

  vendor_account_change:
    detection: vendor.bank_account != previous_payment.bank_account
    action: block_payment
    verification_required:
      method: phone_call
      to: vendor.verified_phone
      by: human
      documentation: signed_confirmation

  amount_discrepancy:
```

```
detection: batch_calculated != batch_expected
action: halt_batch
notification: [ap_supervisor]
investigation_required: true
resume_only_after: human_approval
```

Section 7: Escalation Matrix

Human Version

Situation	Escalate To	Timeframe
Unclear categorization	Senior Bookkeeper	Within 1 hour
Approval needed > $10K	Controller	Within 4 hours
Suspected fraud	Owner + Controller	Immediately
System failure	IT + Supervisor	Within 15 minutes

AI Version

```
escalation:
  low_confidence:
    threshold: confidence < 85%
    escalate_to: senior_bookkeeper
    sla: 1_hour
    include: [transaction_details, ai_analysis, suggested_actions]

  high_value:
    threshold: amount > 10000
    escalate_to: controller
    sla: 4_hours
    require: explicit_approval

  fraud_indicators:
    triggers:
      - duplicate_invoice_different_amount
      - vendor_bank_account_change_recent
      - round_number_just_under_approval_limit
    escalate_to: [owner, controller]
    sla: immediate
    action: block_all_related_transactions

  system_failure:
    triggers:
      - connection_timeout_after_retry
      - authentication_failure
      - data_corruption_detected
    escalate_to: [it_support, ap_supervisor]
```

```
    sla: 15_minutes
    preserve: transaction_state
```

Complete SOP Example: Invoice Entry

SOP-AP-001: Invoice Entry Procedure

```
sop:
  id: AP-INV-ENTRY-v1.0
  title: Invoice Entry Procedure
  category: AP
  ai_enabled: true
  confidence_threshold: 90%

trigger:
  event: invoice_received
  sources: [email, mail, portal]

prerequisites:
  - vendor.exists_in_system
  - user.has_role: [bookkeeper, ap_clerk]

steps:
  - step: 1
    name: Capture Invoice
    human: "Scan or download invoice to staging folder"
    ai:
      action: ocr_extract
      fields: [vendor_name, invoice_number, date, amount, line_items]
      confidence_required: 85%
    validation: all_fields_extracted

  - step: 2
    name: Match Vendor
    human: "Select vendor from dropdown or create new"
    ai:
      action: fuzzy_match
      source: extracted.vendor_name
      target: vendor_master
      threshold: 0.92
    decision:
      - if: match_found
        then: continue
      - if: no_match
        then: escalate(new_vendor_setup)

  - step: 3
```

```yaml
    name: Check Duplicate
    human: "Verify invoice hasn't been entered before"
    ai:
      action: duplicate_check
      keys: [vendor_id, invoice_number, amount, date]
    decision:
      - if: duplicate_found
        then: reject(reason: "Duplicate invoice")
      - if: similar_found
        then: flag_for_review

  - step: 4
    name: Enter Invoice
    human: "Input all invoice details into accounting system"
    ai:
      action: create_bill
      mapping:
        vendor: matched_vendor.id
        invoice_number: extracted.invoice_number
        date: extracted.date
        due_date: calculated(date + vendor.terms)
        amount: extracted.amount
        lines: extracted.line_items
    validation:
      - bill.total == extracted.amount
      - bill.due_date >= today

  - step: 5
    name: Attach Documentation
    human: "Attach original invoice to transaction"
    ai:
      action: attach_document
      document: original_invoice
      to: created_bill
    validation: attachment.exists

  - step: 6
    name: Route for Approval
    human: "Send to appropriate approver based on amount"
    ai:
      action: route_approval
      rules:
        - if: amount <= 1000
          to: ap_supervisor
        - if: amount <= 5000
          to: controller
        - if: amount > 5000
          to: [controller, cfo]
```

```yaml
exceptions:
  ocr_low_confidence:
    trigger: any_field.confidence < 85%
    action: queue_for_manual_review

  vendor_not_found:
    trigger: vendor_match.score < 0.92
    action: escalate(new_vendor_process)

completion:
  log: audit_trail
  notify: submitter
  status: awaiting_approval
```

Conversion Checklist

Use this checklist when converting existing SOPs to AI-infused format:

Analysis Phase

☐ Identify all decision points in current SOP
☐ List all data inputs required
☐ Document validation criteria
☐ Map exception scenarios
☐ Define escalation paths

Conversion Phase

☐ Write machine-readable trigger conditions
☐ Define prerequisite checks
☐ Convert each step to dual format (human + AI)
☐ Create decision trees with conditions
☐ Specify confidence thresholds
☐ Document exception handling rules

Testing Phase

☐ Test AI execution with sample data
☐ Verify escalation triggers work
☐ Confirm validation catches errors
☐ Test exception handling paths
☐ Validate human can override any AI decision

Deployment Phase

☐ Train team on new format
☐ Set initial confidence thresholds conservatively

☐ Monitor AI vs human decisions
☐ Adjust thresholds based on accuracy
☐ Document lessons learned

For more resources: projectbits.com/taxready/ch10

Tax Ready Bookkeeping by Don Lovett | ProjectBits Consulting

Tax Ready(TM) Bookkeeping