### List of Use Cases Implemented by the AP140, AP141, and AP145 RPG Programs

The RPG programs `AP140`, `AP141`, and `AP145`, along with the associated OCL script, implement a single primary use case as part of an IBM System/36 or AS/400 accounts payable (A/P) system for processing employee expense payments. This use case is:

1. **Generate Employee Expense Voucher Selection Spreadsheet and Report**:
   - **Description**: This use case allows the system to process A/P payment transactions for employee expenses, select eligible open payables (vouchers), and produce a detailed report and summary file for payroll integration. It supports multiple payment methods (checks, ACH, wire transfers, employee expenses, utility auto-pay) and handles validations, calculations, and reporting.
   - **Components**:
     - **AP140**: Interactive entry of payment transaction details (company, bank G/L, batch, check date, pay-by date, payment type, vendor/voucher specifics).
     - **AP141**: Matches transactions to open payables and creates payment records with appropriate payment amounts and types.
     - **AP145**: Generates detailed reports and a summary file, validating checks and accumulating totals for reporting.
   - **Inputs**: Company number, bank G/L number, batch number, check date, pay-by date, payment type, vendor/voucher details, and accounting period/year (if 13 periods).
   - **Outputs**: Payment records (`ADPPAY`, `ADPYCK`), detailed reports (`APEEEXP`, `APEEEXPO`), and a summary disk file (`APEEPY`) for payroll integration.

---

### Function Requirement Document

# Employee Expense Processing Function Requirements

## Overview

The **Employee Expense Processing Function** automates the selection, validation, and reporting of accounts payable (A/P) employee expense payments. It processes transactions, matches them to open payables, calculates payment amounts, and generates detailed reports and a payroll summary file. The function supports multiple payment methods (checks, ACH, wire transfers, employee expenses, utility auto-pay) and ensures compliance with accounting rules, including support for 13 accounting periods.

## Inputs

- **Company Number** (`CONO`): Valid company identifier from `APCONT`.
- **Bank G/L Number** (`BKGL`): Valid bank G/L account from `GLMAST` (not deleted/inactive).
- **Batch Number** (`BTCH`): Non-zero batch identifier for grouping payments.
- **Check Date** (`CKDT`): Valid date (MMDDYY) for issuing payments.
- **Pay-By Date** (`DATE`): Optional date (MMDDYY) to filter vouchers by due date.
- **Payment Type** (`KYHOLD`): `' '` (check), `'A'` (ACH), `'W'` (wire), `'E'` (employee expense), `'U'` (utility auto-pay).
- **Vendor Number** (`PTVEND`): Optional vendor identifier from `APVEND` (0 for one-time vendors).
- **Voucher Number** (`PTVO`): Optional voucher identifier from `APOPEN` (0 for whole vendor).

- **Partial Payment Amount** (PTAMT): Optional amount for partial voucher payment.
- **Override Discount** (PTDISC): Optional discount amount to override default.
- **Force Discount** (FDISC): 'D' to force discount, else blank.
- **Pay or Hold** (PTPORH): 'P' to pay, 'H' to hold (for vendor/voucher-specific transactions).
- **Single Check** (PTSNGL): 'S' for single check per vendor, else blank.
- **Prepaid Flag** (PTMKPP): 'P', 'A', 'W', 'E', 'U' for prepaid vouchers, else blank.
- **Prepaid Check Number** (PTPPCK): Check number for prepaid vouchers.
- **Prepaid Date** (PTPPDT): Date for prepaid vouchers.
- **Period/Year** (KYPD, KYPDYY): Accounting period (1–13) and year (if 13 periods enabled in GSCONT).

## Outputs

- **Payment Records** (ADPPAY): Records with company, vendor, voucher, payment amount, discount, check number, and payment type.
- **Check Records** (ADPYCK): Check details with check number, amount, and status (normal, prepaid, credit/no pay, full stub).
- **Reports** (APEEEXP, APEEEXPO): Detailed reports with company, vendor, invoice details, check totals, and company totals.
- **Summary File** (APEEPY): Disk file with vendor payroll ID and payment amounts for payroll integration.

## Process Steps

1. **Validate Inputs**:
    - Verify CONO exists in APCONT.
    - Ensure BKGL is valid in GLMAST (not deleted/inactive).
    - Confirm BTCH ≠ 0.
    - Validate CKDT and DATE (MMDDYY format, valid month/day, leap year).
    - If 13 accounting periods enabled (GX13GL = 'Y' in GSCONT), ensure KYPD is 1–13 and CKDT falls within period dates in GSTABL.
    - Validate KYHOLD is ' ', 'A', 'W', 'E', or 'U'.
    - For vendor-specific transactions, verify PTVEND exists in APVEND and PTVO in APOPEN (if provided).
2. **Create Transactions**:
    - Store transaction details in ADPYTR with sequence number, company, bank G/L, batch, check date, pay-by date, payment type, and vendor/voucher details.
3. **Match Open Payables**:
    - For pay-by-date transactions (DATE ≠ 0):
        - Select APOPEN records where OPCONO = CONO, OPBKGL = BKGL, due date (OPDUED) ≤ DATE, and not deleted (OPDEL ≠ 'D') or halted (OPHALT ≠ 'H').
        - Match payment type: ' ' (non-ACH/wire/employee/utility), 'A' (OPHALT = 'A'), etc.
    - For vendor-specific transactions:
        - Select APOPEN records matching PTVEND (and PTVO if provided), OPCONO, and OPBKGL (if whole vendor).
        - Override hold (OPHALT = 'H') if PTPORH = 'P'.
        - Validate prepaid flags match KYHOLD.

4. **Calculate Payment Amounts**:
    - Gross amount: `OPGRAM` from `APOPEN`.
    - Discount: Apply `PTDISC` (if provided), else `OPDISC` from `APOPEN`. Set to 0 if voucher is past due (`OPDUED > CKDT`) or partially paid (`OPPPTD ≠ 0`) and `FDISC ≠ 'D'`.
    - Payment amount: `OPLPAM = OPGRAM - OPDISC - OPPPTD`.
    - Partial payment: If `PTAMT ≠ 0`, set `OPLPAM = PTAMT` and adjust remaining `PTAMT`.
5. **Assign Check Numbers**:
    - Use `PTPPCK` for prepaid vouchers.
    - Use next check number (`PTNXCK`) from `ADPYTR` for non-prepaid.
    - Set check number to 0 for credit/no pay (`OPLPAM = 0`).
    - Increment `PTNXCK` for each new check unless full stub or credit/no pay.
6. **Validate Checks**:
    - Ensure non-void checks do not exist in `APCHKR` or are not open (`AMCODE ≠ 'O'`).
    - For void checks, ensure they exist, are open, and match the full amount.
    - Mark negative or zero-amount checks as credit/no pay ( `Hypothesized: (`AXRECD = 'C'`).
7. **Generate Outputs**:
    - Write payment records to `ADPPAY` with company, vendor, voucher, payment amount, discount, check number, payment type, and single check flag.
    - Write check records to `ADPYCK` with check number, amount, and status (normal, prepaid, credit/no pay, full stub).
    - Generate reports (`APEEEXP`, `APEEEXPO`) with:
        - Headers: Company name, payment type, date, time.
        - Details: Sequence number, invoice number, description, gross amount, discount, partial paid to date, payment amount, due date, vendor, voucher number.
        - Totals: Check totals, company totals (employee count, gross, discount, payment amounts).
    - Write summary file (`APEEPY`) with vendor payroll ID (`VNPRID`) and negative payment amount.

# Business Rules

1. **Validation**:
    - Company, bank G/L, and batch must be valid and non-zero.
    - Dates must be valid and align with accounting periods (if 13 periods).
    - Payment type must match voucher type in `APOPEN`.
    - Vendor/voucher must exist for specific transactions.
2. **Payment Selection**:
    - Pay-by-date: Select vouchers due by `DATE`, not on hold unless overridden.
    - Vendor-specific: Match vendor (and voucher if specified), override hold if `PTPORH = 'P'`.
    - Prepaid vouchers must match payment type (`OPPAID = KYHOLD`).
3. **Discounts**:
    - Apply override discount (`PTDISC`) or default (`OPDISC`).
    - Set discount to 0 for past due or partially paid vouchers unless forced (`FDISC = 'D'`).
4. **Payment Amount**:

    - Calculate as `OPGRAM - OPDISC - OPPPTD`.
    - Adjust for partial payments (`PTAMT`).
5. **Check Handling**:
    - Single checks (`OPSNGL = 'S'`) for one-time vendors or specified cases.
    - Maximum 12 invoices per stub; mark as full stub (`AXRECD = 'F'` or `'V'`).
    - Negative/zero-amount checks marked as credit/no pay.
6. **Reporting**:
    - Include vendor name from `APVEND` or `APOPEN`.
    - Display payment type labels (e.g., "PAY BY CHECK", "PAY BY UTIL-AUPY").
    - Report errors for invalid checks (e.g., "CHECK IS ALREADY OPEN").

## Calculations

- **Payment Amount**: `OPLPAM = OPGRAM - OPDISC - OPPPTD`. If `PTAMT ≠ 0`, `OPLPAM = min(PTAMT, OPGRAM - OPDISC - OPPPTD)` and update `PTAMT`.
- **Discount**: `OPDISC = PTDISC` (if provided) or `OPOPEN.OPDISC`. Set to 0 if past due (`OPDUED > CKDT`) or `OPPPTD ≠ 0` and `FDISC ≠ 'D'`.
- **Check Number**: `THISCK = OPCKNO` (prepaid), `PTNXCK` (non-prepaid), or 0 (credit/no pay). Increment `PTNXCK` unless full stub or credit/no pay.
- **Totals**: Accumulate gross (`CKGRAM`, `C6GRAM`, `P6GRAM`, `L6GRAM`), discount (`CKDISC`, `C6DISC`, `P6DISC`, `L6DISC`), and payment (`CKAMT`, `C6LPAM`, `P6LPAM`, `L6LPAM`) at check and company levels.

## Dependencies

- **Files**:
    - `APCONT`: Company data.
    - `GLMAST`: G/L accounts.
    - `GSCONT`: System settings (13 periods).
    - `GSTABL`: Period end dates.
    - `APVEND`: Vendor details.
    - `APOPEN`: Open payables.
    - `APCHKR`: Check register.
    - `ADPYTR`: Transaction input.
    - `ADPPAY`: Payment output.
    - `ADPYCK`: Check output.
    - `APEEPY`: Payroll summary output.
    - `APEEEXP`, `APEEEXPO`: Report output.