

The provided document is an AS/400 Operation Control Language (OCL) procedure ( `AP200.oc136.txt` ) that manages the **Purchase Journal** for posting voucher transactions in an Accounts Payable (A/P) system. Below, I'll explain the process steps in a clear, structured manner, breaking down the key operations, checks, and logic flow of the OCL file.

## Overview

The `AP200` OCL procedure automates the posting of A/P voucher or wire transfer transactions to the Purchase Journal. It performs validations, sorts data, updates files, and generates reports. The procedure includes error handling, file management, and integration with other processes (e.g., inventory, job costing). It also supports wire transfer (`WT`) processing and ensures that certain conditions are met before proceeding.

## Process Steps

### 1. Initial Setup and Metadata

- **Comments and Revisions:**
  - The file includes metadata about revisions, such as:
    - `JB01` (09/14/14, Jan Beccari): Added posting to inventory transaction holding ( `INTZH` ).
    - `JK01` (03/27/15, Jimmy Krajacic): Added support for carrier freight invoices ( `FRGINV` ) when a voucher is deleted.
    - `JB` (06/07/21): Prevents A/P voucher posting during inventory beginning-of-week processing ( `INTSZZ` ).
- **Wire Transfer Check:**
  - The procedure checks if the user selected the "Wire Transfer" journal option ( `WT` ).
    - If `?3?/WT`, sets `P20='APWT?WS?'` (wire transfer file).
    - Otherwise, sets `P20='APTR?WS?'` (standard transaction file).
- **Switch Initialization:**
  - Initializes `SWITCH` to `00000000` for conditional logic.
  - Sets `SWITCH 10000000` if specific file conditions are met (e.g., `DATAF1-?9??20?` or `?F'A,?9??20?'?/00000000`).

### 2. Pre-Posting Validations

The procedure performs several checks to ensure the system is in a valid state for posting:

- **No Voucher or Wire Transaction File:**
  - If `SWITCH1-1` (indicating no A/P voucher or wire transaction file exists):
    - Displays: *"NO A/P VOUCHER OR WIRE TRANSACTION FILE TO POST"*.
    - Prompts user to press `0`, `ENTER` to cancel.
    - Jumps to `END`.
- **Payment Cycle Conflict:**
  - If `DATAF1-?9?APPT?WS?` (payment cycle file exists):
    - Deletes `?9?APPT?WS?,F1` if conditions allow.
    - Displays a warning: *"THE PAYMENT CYCLE MUST BE ENDED FROM THIS WORKSTATION BEFORE THIS PURCHASE JOURNAL CAN BE RUN"*.
    - Prompts user to cancel and jumps to `END`.
- **Concurrent Purchase Register:**
  - If `ACTIVE-AP200` (another A/P purchase register is running):
    - Displays: *"AN A/P PURCHASE REGISTER IS ALREADY IN PROGRESS. PLEASE TRY AGAIN IN A FEW MINUTES"*.
    - Prompts user to cancel and jumps to `END`.
- **Inventory Beginning-of-Week Conflict**
  - If `DATAF1-?9?INTSZZ` (inventory beginning-of-week process is active):
    - Displays: *"INVENTORY BEGINNING OF WEEK IS IN PROGRESS. PLEASE TRY AGAIN IN A FEW MINUTES"*.
    - Prompts user to cancel and jumps to `END`.
- **Voucher Batch Errors:**
  - Checks `?9?APSTAT` for errors in the voucher table (from `AP110`).
  - If `?L'231,3'?/YES` (errors exist):
    - Displays: *"ERRORS EXIST IN THE BATCH. PLEASE RETURN TO THE BATCH AND CORRECT THE ERRORS"*.
    - Pauses and jumps to `END`.

### 3. File and Variable Initialization

- **Clear Variables:**
  - Sets `LOCAL BLANK-*ALL` to clear local variables.
- **Set Journal Type:**
  - If `?3?/WT`, sets `OFFSET-198,DATA-'WT*** WIRE TRANSFER ***'`.
  - Otherwise, sets `OFFSET-198,DATA-' '`.
- **Set Workstation:**
  - Sets `OFFSET-300,DATA-'?WS?'` (workstation ID).
- **Create Temporary File:**
  - Builds `?9?APPT?WS?` (payment transaction file) with 200 records, 128 bytes each.

#### 4. Load and Run Initial Program (AP200P)

- **Program:** AP200P
- **Files:**
  - APTRAN : Transaction file ( ?9??20? ).
  - APPYTR : Payment transaction file ( ?9?APPT?WS? ).
  - GSTABL , GSCONT , GLCONT : Shared general ledger and system control files.
- **Action:**
  - Runs AP200P to process initial transaction data.
- **Cancel Check:**
  - If ?L'129,6' ?/CANCEL , deletes APPT?WS? and jumps to END .

#### 5. File Cleanup and Temporary File Creation

- **Delete Temporary Files:**
  - Deletes APPJ?WS? , APPK?WS? , APJC?WS? using GSDELETE .
- **Create Job Cost Transaction File:**
  - Builds ?9?APJC?WS? with 999,000 records, 128 bytes each.

#### 6. Sort Transactions (#GSORT)

- **Program:** #GSORT
- **Input File:** ?9??20? (transaction file).
- **Output File:** ?9?APXX?WS? (sorted transaction file, 999,000 records).
- **Sort Criteria:**
  - Sorts by:
    - Company ( FNC 2 3 ).
    - Vendor ( FNC 12 16 ).
    - Entry/Entry Sequence ( FNC 4 11 ).
  - Copies fields 1–256 and 257–404 ( FDC ).
- **Action:**
  - Executes sort to organize transactions for processing.

#### 7. Run Main Purchase Journal Program (AP200)

- **Program:** AP200
- **Files:**
  - APTRAN : Sorted transactions ( ?9?APXX?WS? ).
  - APCONT , APVEND , APOpen , APOpenH , APOpenD , APOpenV : A/P control, vendor, and open item files (shared).
  - APHISTH , APHISTD , APHISTV : A/P history files (shared).
  - APINVH : Invoice header file (shared).
  - POFILEH , POFILED : Purchase order files (shared).
  - JCTran : Job cost transactions ( ?9?APJC?WS? ).
  - APPJJR : Journal register ( ?9?APPJ?WS? ).
  - APPYTR : Payment transactions ( ?9?APPT?WS? ).
  - FRCINH , FRCFBH : Freight invoice files (shared).
- **Printer Overrides:**
  - If ?9?/G , sets output queue to QUSRSYS/APPOST .
  - Otherwise, sets output queue to QUSRSYS/TESTOUTQ .
- **Action:**
  - Runs AP200 to process transactions, update A/P files, and generate the Purchase Journal report.

#### 8. Sort Journal Register (#GSORT)

- **Program:** #GSORT
- **Input File:** ?9?APPJ?WS? (journal register).
- **Output File:** ?9?APPK?WS? (sorted journal register, 999,000 records).
- **Sort Criteria:**
  - Sorts by:
    - Company ( FNC 2 3 ).
    - Control/Distribution ( FNC 12 12 ).
    - A/P, Expense, Inter-Company ( FNC 106 115 ).
    - G/L Account ( FNC 13 20 ).
  - Includes records where field 1 is not empty ( I C 1 1NECD ).
- **Action:**
  - Sorts the journal register for further processing.

#### 9. Run Journal Summary Program (AP205)

- **Program:** AP205
- **Files:**
  - APPJJR : Journal register ( ?9?APPJ?WS? ).
  - AP205S : Sorted journal register ( ?9?APPK?WS? ).
  - APCONT : A/P control file (shared).
  - TEMGEN : Temporary general ledger file (shared).
- **Printer Overrides:**

- Same as AP200 (output queue APPOST or TESTOUTQ).
- **Action:**
  - Runs AP205 to summarize journal entries and produce reports.

## 10. Post Job Cost Transactions (Conditional)

- **Condition:**
  - If ?F'A, ?9?APJC?WS?'?/00000000 (job cost transaction file exists) and procedure JC200 exists in the library ( ?CLIB? ).
- **Action:**
  - Calls JC200 with APJC?WS? to post job cost transactions.

## 11. Post A/P Invoices to Inventory Receipts (AP210)

- **Program:** AP210
- **Files:**
  - APTRAN : Transaction file ( ?9??20? ).
  - INFIL1, INTZH1 : Inventory files (shared).
- **Printer:**
  - Output to APLIST (device PJ, form JBAP, priority 0).
- **Action:**
  - Posts A/P invoices to inventory receipt records.

## 12. Cleanup Temporary Files

- **Action:**
  - Deletes temporary files using GSDELETE :
    - APPJ?WS?, APPK?WS?, APTX?WS?, APXX?WS? .
    - ?20?, APCT?WS?, APJC?WS? .
    - APPT?WS? (if no records exist).
- **Condition:**
  - If DATAF1-?9?APPT?WS?, skips deletion and jumps to END .

## 13. Automatic Prepaid Invoice Processing

- **Action:**
  - Displays: "AUTOMATIC PROCESSING OF PREPAID INVOICES IS EXECUTING".
  - Calls procedures:
    - AP150 (auto mode, parameter ?3? ).
    - AP160 (auto mode, parameter ?3? ).
    - AP250 (auto mode, parameter ?3? ).
- **Purpose:**
  - Processes prepaid invoices automatically.

## 14. Final Cleanup

- **LMS Identifier Deletion:**
  - If DATAF1-?9?LMS?WS? (LMS batch identifier exists), deletes LMS?WS? .
- **Reset State:**
  - Clears all local variables ( LOCAL BLANK-\*ALL ).
  - Resets SWITCH to 00000000 .

## 15. End of Procedure

- **Tag:** END
- **Action:**
  - Terminates the procedure after all processing or upon cancellation/error.

---

## Key Features and Notes

- **Error Handling:**
  - The procedure includes robust checks to prevent conflicts (e.g., concurrent processes, payment cycle locks, inventory conflicts).
  - User prompts ensure manual intervention when errors occur.
- **File Management:**
  - Temporary files ( APPJ?WS?, APPK?WS?, etc.) are created and deleted to manage data during processing.
  - Shared files (e.g., APCONT, APVEND) are accessed in shared mode ( DISP-SHR ) to allow concurrent access.
- **Sorting:**
  - Two sorting steps ( #GSORT ) organize transactions and journal entries for accurate posting and reporting.
- **Modularity:**
  - Calls external programs ( AP200P, AP200, AP205, AP210, JC200 ) and procedures ( AP150, AP160, AP250 ) for specific tasks.
- **Wire Transfer Support:**
  - Differentiates between standard transactions ( APTR?WS? ) and wire transfers ( APWT?WS? ).
- **Inventory Integration:**
  - Posts to inventory receipts ( AP210 ) and prevents conflicts with inventory processes ( INTSZZ ).
- **Freight Invoices:**
  - Supports carrier freight invoice processing ( FRCINV ) when vouchers are deleted.

---

## Flow Summary

1. **Validate environment** (no conflicts, no errors in voucher batch).
  2. **Initialize variables and files** (set `WT`, create `APPT?WS?` ).
  3. **Process initial transactions** ( `AP200P` ).
  4. **Sort transactions** ( `#GSORT` to `APXX?WS?` ).
  5. **Post to Purchase Journal** ( `AP200`, update A/P and related files).
  6. **Sort journal register** ( `#GSORT` to `APPK?WS?` ).
  7. **Summarize journal** ( `AP205` ).
  8. **Post job cost transactions** (if applicable, `JC200` ).
  9. **Post to inventory receipts** ( `AP210` ).
  10. **Process prepaid invoices** ( `AP150`, `AP160`, `AP250` ).
  11. **Clean up** (delete temporary files, reset state).
- 

## Assumptions and Clarifications

- **Parameters:**
  - `?9?` : Library name (dynamic).
  - `?WS?` : Workstation ID.
  - `?3?` : Likely a mode or batch parameter.
  - `?20?` : File name ( `APTR?WS?` or `APWT?WS?` based on `WT` ).
- **System Context:**
  - Runs on IBM AS/400 with RPG programs and OCL.
  - Assumes a multi-user environment with shared files.
- **External Dependencies:**
  - Programs: `AP200P`, `AP200`, `AP205`, `AP210`, `#GSORT`, `JC200` .
  - Procedures: `AP150`, `AP160`, `AP250` .
  - Files: A/P, inventory, purchase order, job cost, and freight invoice files.

If you need further details on specific programs, file structures, or logic, let me know!