The provided document is an Operations Control Language (OCL) program, `AP100.ocl36.txt`, used in IBM midrange systems (e.g., AS/400 or iSeries) for Accounts Payable (A/P) voucher entry and editing. Below is an explanation of the process steps, followed by a list of external programs called and tables (files) used.

Process Steps of the RPG .OCL Program

The OCL program orchestrates the A/P voucher entry and edit process, handling validation, file management, and user interaction for regular A/P transactions, freight invoices from the LMS system, or other entry methods. Here's a breakdown of the steps:

1. **Initial Validation and Inventory Check**:

   - The program checks if an inventory process (`INTSZZ`) is in progress by evaluating `DATAF1-?9? INTSZZ`.
   - If true, it displays messages indicating that the inventory process is ongoing, instructs the user to try again later, pauses for user input (press 0 to cancel), and branches to the `END` tag, halting execution.
   - This prevents A/P voucher posting during specific inventory processes to avoid data conflicts.

2. **Wire Transfer (WT) Handling**:

   - The program checks if the transaction involves a wire transfer (`?3?/WT`).
   - If true, it sets `P20` to `'APWT?WS?'` and adds a local data field at offset 198 with `'WT*** WIRE TRANSFER ***'`.
   - If false, it sets `P20` to `'APTR?WS?'` and clears the data field at offset 198.
   - This distinguishes wire transfer transactions from regular A/P transactions.

3. **LMS Freight Invoice vs. Regular A/P Check**:

   - The program determines whether the process is for freight invoices imported from the LMS system or regular A/P transactions.
   - **Rule**: If an A/P transaction file (`?9??20?`) exists, LMS processing is not allowed.
   - If the A/P transaction file does not exist, the program displays a selection screen (`AP100S`) for the user to choose between LMS automatic processing or regular A/P entry.
   - If the LMS file (`DATAF1-?9?LMS?WS?`) exists, the program deletes the A/P transaction file (`APTX?WS?`) using `GSDELETE`.

4. **User Selection for Entry Method**:

   - The program loads the `AP100S` screen to allow the user to select the entry method:
     - **ARGLMS**: Triggers the `AP125` program for LMS processing (with parameter `N`) and branches to `END`.
     - **PAPER**: Triggers the `AP125` program for paper-based A/P entry (with parameter `P`) and branches to `END`.
     - **FLEXI**: Triggers the `AP106` program for flexible entry and branches to `END`.
     - **CANCEL**: If the user cancels (`?L'129,6'?/CANCEL`), the program branches to `END`.
   - If the A/P transaction file exists (`?F'A,?9?APTR?WS?'?/00000000`) or `DATAF1-?9??20?` is true, the program skips to the `AROUND` tag, bypassing the selection screen.

5. **File and Index Creation**:

   - If the A/P transaction file (`?9??20?`) does not exist, the program creates it using `BLDFILE` with 500 records, 404 bytes each, and specific key fields.
   - It also builds an index for the A/P transaction file (`?9?APTX?WS?`) with keys at positions 12, 5, 385, and 20.
   - For the inventory transaction holding file (`INTZH1`), it builds an index if it exists (`DATAF1-?9?INTZH1`).
   - If the job cost file (`JCCOST`) exists, it creates a file with 999,000 records, 256 bytes each.

6. **File Definitions for AP100**:

   - The program defines multiple files for the `AP100` program, including `APTRAN`, `APTRANX`, `APTRNX`, `APOPEN`, `APOPENH`, `APOPNHC`, `APCONT`, `APDATE`, `APVEND`, `APVENDX`, `GLMAST`, `JCCOST`, `GSTABL`, `POFILE`, `POADDR`, `APINVH`, `APHSTHC`, `INFIL1`, `INTZH1`, `POPROJ`, `SA5FIUD`, `SA5FIUM`, `SA5MOUD`, `SA5MOUM`, `BICONT`, `GSCTUM`, `FRCINH`, and `FRCFBH`.
   - These files are opened with shared access (`DISP-SHR`) and extended as needed (e.g., `EXTEND-100` for `APTRAN`).
   - The program then runs `AP100` for voucher entry.

7. **Post-Entry Processing**:

   - If the A/P transaction file is empty (`?F'A,?9??20?'?/00000000`), the program branches to `END`.
   - It sets local variables at offsets 135 (`?WS?`) and 221 (`?9??20?`).
   - It deletes and rebuilds the A/P check transaction file (`APCT?WS?`) with 500 records, 80 bytes each.

8. **Voucher Transaction Edit (AP110)**:

   - The program loads `AP110` for editing A/P voucher transactions.
   - It defines files like `APTRAN`, `APCONT`, `APCHKR`, `APCHKT`, `APTRNX`, `GLMAST`, `APOPNHC`, `APINVH`, `GSTABL`, `APSTAT`, `APVEND`, `INFIL1`, and `INTZH1`.
   - It overrides the printer file (`APLIST`) to output to `QUSRSYS/APEDIT` or `QUSRSYS/TESTOUTQ` based on the `?9?/G` condition.
   - The program runs `AP110` to edit transactions.

9. **Prepaid Check Edit (AP115)**:

   - If the LMS file (`DATAF1-?9?LMS?WS?`) exists or the A/P check transaction file (`APCT?WS?`) is empty, the program skips to `NOPPD`.
   - Otherwise, it loads `AP115` for prepaid check editing, defining files like `APCHKT`, `APCHKTX`, `APCHKR`, and `APCONT`.
   - It overrides the printer file (`APLIST`) similarly and runs `AP115`.

10. **Purchase Order Edit (Skipped)**:

    - The program explicitly skips the purchase order edit section by branching to `END`.
    - If executed, it would:
      - Sort the A/P transaction file (`?9??20?`) into `?9?APPO?WS?` using `#GSORT` with specific sort criteria (company, P/O number, entry sequence).

- Load `AP120` for A/P and purchase order voucher editing, using files like `APTRAN`, `APTRANH`, `APCONT`, and `POFILE`.

11. **Program Termination**:

    - The program clears all local variables (`LOCAL BLANK-*ALL`) and ends execution at the `END` tag.

External Programs Called

The OCL program invokes the following external programs:

1. **AP100S**: Displays the selection screen for choosing the entry method (LMS, paper, or flexible).
2. **AP100**: Handles A/P voucher entry.
3. **AP106**: Processes flexible A/P entry (triggered by `FLEXI` selection).
4. **AP110**: Edits A/P voucher transactions.
5. **AP115**: Edits prepaid checks.
6. **AP120**: Handles A/P and purchase order voucher editing (skipped in this execution).
7. **AP125**: Processes LMS or paper-based A/P entry (triggered by `ARGLMS` or `PAPER` selection).
8. **#GSORT**: Sorts the A/P transaction file for purchase order editing (skipped).

Tables (Files) Used

The program references the following files (tables), primarily opened with shared access (`DISP-SHR`):

1. **APTRAN**: A/P transaction file (`?9??20?`).
2. **APTRANX**: Alternate A/P transaction file (`?9??20?`).
3. **APTRNX**: A/P transaction index file (`?9?APTX?WS?`).
4. **APOPEN**: A/P open items file (`?9?APOPEN`).
5. **APOPENH**: A/P open items history file (`?9?APOPNH`).
6. **APOPNHC**: A/P open items history control file (`?9?APOPNHC`).
7. **APCONT**: A/P control file (`?9?APCONT`).
8. **APDATE**: A/P date file (`?9?APDATE`).
9. **APVEND**: A/P vendor file (`?9?APVEND`).
10. **APVENDX**: A/P vendor index file (`?9?APVENX`).
11. **GLMAST**: General ledger master file (`?9?GLMAST`).
12. **JCCOST**: Job cost file (`?9?JCCOST`).
13. **GSTABL**: General system table (`?9?GSTABL`).
14. **POFILE**: Purchase order file (`?9?POFILE`).
15. **POADDR**: Purchase order address file (`?9?POADDR`).
16. **APINVH**: A/P invoice history file (`?9?APINVH`).
17. **APHSTHC**: A/P history control file (`?9?APHSTHC`).
18. **INFIL1**: Inventory file 1 (`?9?INFIL1`).
19. **INTZH1**: Inventory transaction holding file (`?9?INTZH1`).
20. **POPROJ**: Purchase order project file (`?9?POPROJ`).
21. **SA5FIUD**: Sales file (unknown purpose, possibly related to financial updates) (`?9?SA5FIUD`).
22. **SA5FIUM**: Sales file (unknown purpose, possibly related to financial updates) (`?9?SA5FIUM`).
23. **SA5MOUD**: Sales file (unknown purpose, possibly related to monthly updates) (`?9?SA5MOUD`).

24. **SA5MOUM**: Sales file (unknown purpose, possibly related to monthly updates) (`?9?SA5MOUM`).
25. **BICONT**: Business intelligence control file (`?9?BICONT`).
26. **GSCTUM**: General system control table (unknown purpose) (`?9?GSCTUM`).
27. **FRCINH**: Freight invoice history file (`?9?FRCINH`).
28. **FRCFBH**: Freight feedback history file (`?9?FRCFBH`).
29. **APCHKR**: A/P check register file (`?9?APCHKR`).
30. **APCHKT**: A/P check transaction file (`?9?APCT?WS?`).
31. **APCHKTX**: Alternate A/P check transaction file (`?9?APCT?WS?`).
32. **APSTAT**: A/P status file (`?9?APSTAT`).
33. **APPO?WS?**: Sorted A/P purchase order file (used by `#GSORT`, temporary).

Summary

The `AP100.ocl36.txt` OCL program manages the A/P voucher entry and edit process, ensuring no conflicts with inventory processes, handling wire transfers, and allowing user selection between LMS freight invoices, paper-based, or flexible A/P entry. It creates and indexes necessary files, runs appropriate programs (`AP100`, `AP106`, `AP110`, `AP115`, `AP125`, `#GSORT`), and skips purchase order editing. The program uses a comprehensive set of files to store transaction, vendor, inventory, and financial data, ensuring proper integration with the broader A/P system.