Below is a Functional Requirements Document (FRD) for the RPG programs `AP1011` and `AP1012`, which are designed to create Accounts Payable (A/P) voucher entry detail totals by prorating carrier freight based on sales details or miscellaneous freight amounts. Since both programs are intended to perform the same function with minor differences, this FRD consolidates the requirements for a single system, noting any variations where applicable. The document is structured to provide a clear, comprehensive description of the system's purpose, functionality, data requirements, and constraints.

Functional Requirements Document: A/P Voucher Freight Allocation System

1. Document Information

- **Document Title**: Functional Requirements Document for A/P Voucher Freight Allocation System
- **Document ID**: FRD-AP1011-AP1012-001
- **Version**: 1.0
- **Date**: July 01, 2025
- **Prepared By**: Grok 3, xAI
- **Purpose**: To define the functional requirements for the A/P Voucher Freight Allocation System, implemented by RPG programs `AP1011` and `AP1012`, which prorate carrier freight costs across sales detail or miscellaneous lines for Accounts Payable voucher entry.

2. System Overview

The A/P Voucher Freight Allocation System processes sales invoice data to allocate carrier freight costs across detail or miscellaneous lines based on net gallons or freight amounts, respectively. The system generates or updates A/P transaction records in the `APTRAN` file, ensuring accurate freight cost distribution and proper General Ledger (G/L) account assignment. The system supports both regular sales invoices (with detail lines) and miscellaneous-only invoices (without detail lines).

2.1 Objectives

- Calculate total net gallons for freight allocation from sales detail records.
- Prorate freight costs across detail lines based on net gallons or miscellaneous lines based on freight amounts.
- Assign appropriate freight G/L accounts based on product codes or miscellaneous types.
- Create or update A/P transaction records with prorated freight amounts, discounts, and relevant details.
- Ensure data integrity by filtering records within one year of the invoice date.

2.2 Scope

- **In-Scope**:
  - Processing sales detail records from `SA5FIUD` or `SA5MOUD` and miscellaneous records from `SA5FIUM` or `SA5MOUM`.
  - Retrieving freight G/L accounts from `GSCTUM`, `GSTABL`, or `BICONT`.
  - Writing or updating A/P transaction records in `APTRAN`.
  - Handling discounts and ensuring total freight allocation matches the input freight amount.

- **Out-of-Scope**:
  - Generation of sales invoices or modification of source data in `SA5FI*` or `SA5MO*` files.
  - Validation of input parameters beyond what is provided in the `SALES` data structure.
  - Integration with external systems beyond file I/O.

3. Functional Requirements

3.1 Input Processing

- **FR-001**: The system shall accept input parameters via a `SALES` data structure with the following fields:
  - `SACO` (2 bytes): Company number.
  - `SAORD` (6 bytes): Order number.
  - `SASRN#` (3 bytes): Shipping reference number.
  - `SASEQ` (3 bytes): Sequence number.
  - `FRTTOT` (7,2): Total freight amount to allocate.
  - `VEND` (5 bytes): Vendor number.
  - `ENTNUM` (5 bytes): Entry number.
  - `EXGL` (8 bytes): Expense G/L number.
  - `DSPC` (4,3): Discount percentage.
  - `CMPDT8` (8 bytes): Comparison date (YYYYMMDD) for filtering records.
  - `S@FIMO` (1 byte): Table indicator (`F` for `SA5FI*`, `M` for `SA5MO*`).
  - `S@DM` (1 byte): Record type (`D` for detail, `M` for miscellaneous).
- **FR-002**: If `S@FIMO` is blank, the system shall determine the appropriate table (`SA5FI*` or `SA5MO*`) and record type (`D` or `M`) by searching for valid records in `SA5FIUD`, `SA5MOUD`, `SA5FIUM`, or `SA5MOUM` matching `SACO`, `SAORD`, `SASRN#`, and `CMPDT8` (specific to `AP1012`).

3.2 Data Retrieval and Filtering

- **FR-003**: The system shall read detail records from `SA5FIUD` (sales detail) or `SA5MOUD` (move detail) when `S@FIMO = 'F'` or `S@FIMO = 'M'`, respectively, filtering by:
  - `S5CO# = SACO` (company number).
  - `S5ORD# = SAORD` (order number).
  - `S5SRN# = SASRN#` (shipping reference number).
  - `S5SHD8 >= CMPDT8` (ship date within one year of invoice date).
- **FR-004**: The system shall read miscellaneous records from `SA5FIUM` (sales miscellaneous) or `SA5MOUM` (move miscellaneous) when no detail records are found, filtering by:
  - `SMCO# = SACO` (company number).
  - `SMORD# = SAORD` (order number).
  - `SMSRN# = SASRN#` (shipping reference number).
  - `SMSHD8 >= CMPDT8` (ship date within one year of invoice date).
  - `SMMSTY = 'F'` (freight type).
  - `SMGLNO ≠ 0` (non-zero G/L number).
- **FR-005**: The system shall retrieve freight G/L numbers (`FEGL`) as follows:
  - For detail records with alpha characters in `S5PROD` (product code), use `CUFEGL` from `GSCTUM` based on `S5CO#`, `S5PROD`, `S5CNTR`, and `S5UM`.
  - For detail records with numeric `S5PROD`, combine `TBFEG4` (first 4 digits) from `GSTABL` (keyed by `CNTRPF` and `S5TANK`) with `S5PROD`, or use `BCFRGL` from `BICONT` if not found.

- For miscellaneous records, use `SMGLNO` directly.

3.3 Freight Proration

- **FR-006**: For detail records:
  - Calculate total net gallons (`TTLQTY`) by summing `S5NGAL` from filtered records.
  - Calculate the percentage (`PCTHLD`) for each record as `S5NGAL / TTLQTY`.
  - Multiply `PCTHLD` by `FRTTOT` to compute the freight amount (`AMTITM`) for each line.
  - Adjust the last record's amount to ensure the sum of all amounts equals `FRTTOT`:
    - If sum equals `FRTTOT`, no adjustment.
    - If sum is less than `FRTTOT`, add the difference to the last record.
    - If sum is greater than `FRTTOT`, subtract the difference from the last record.
- **FR-007**: For miscellaneous records (when no detail records exist):
  - Calculate total miscellaneous freight (`TTLMFT`) by summing `SMMAMT * SMMQTY` for filtered records.
  - Calculate the percentage (`PCTHLD`) for each record as `(SMMAMT * SMMQTY) / TTLMFT`.
  - Multiply `PCTHLD` by `FRTTOT` to compute the freight amount (`FRTAMT`) for each line.
  - For the last record, set `FRTAMT = FRTTOT - sum(previous FRTAMT)` to ensure the total matches `FRTTOT`.

3.4 A/P Transaction Processing

- **FR-008**: The system shall write or update records in `APTRAN` (A/P transaction file) as follows:
  - **Key Fields**: `S5CO#` or `SMCO#` (company number), `ENTNUM` (entry number), `COUNT2` or `COUNT3` (sequence number).
  - **Common Fields**:
    - Record type: `A` (active).
    - Vendor number: `VEND`.
    - Freight G/L: `FEGL`.
    - Description: `FRTCHG` (detail) or `MISC CHARGE` (miscellaneous) from `MSG` array.
    - Order number: `SAORD` or `SMORD#`.
    - Shipping reference number: `SASRN#` or `SMSRN#`.
    - Discount percentage: `DSPC`.
    - Status: `C` (closed).
    - Zero-filled fields for unused amounts (e.g., `ATDISC`, `ATQTY`, `ATGALN`, etc.).
  - **Detail-Specific Fields**:
    - Product code: `S5PROD`.
    - Container code: `S5CNTR`.
    - Amount: `LINAMT` (prorated freight).
  - **Miscellaneous-Specific Fields**:
    - Amount: `FRTAMT` (adjusted for last record).
- **FR-009**: If an `APTRAN` record exists for the key (`CHAIN` succeeds), update it (`UPDT` or `UPDTM`); otherwise, write a new record (`ADDT` or `ADDTM`).

3.5 Error Handling and Data Validation

- **FR-010**: The system shall skip records marked as deleted (`S5DEL = 'D'` or `SMDEL = 'D'`).

- **FR-011**: If no valid records are found in SA5FIUD/SA5MOUD or SA5FIUM/SA5MOUM, the system shall exit without writing to APTRAN.
- **FR-012**: If TTLQTY = 0 for detail records or TTLMFT = 0 for miscellaneous records, set PCTHLD to zero to avoid division by zero.

4. Data Requirements

4.1 Input Files

- **SA5FIUD** (Sales Detail, 1024 bytes, keyed):
  - Fields: S5DEL, S5CO#, S5CUST, S5SHIP, S5PROD, PRD1-4, S5TANK, S5ORD#, S5UM, S5NGAL, S5CNTR, S5SHD8, S5SRN#.
- **SA5MOUD** (Move Detail, 1024 bytes, keyed):
  - Same fields as SA5FIUD.
- **SA5FIUM** (Sales Miscellaneous, 1024 bytes, keyed):
  - Fields: SMDEL, SMCO#, SMCUST, SMSHIP, SMMQTY, SMMAMT, SMORD#, SMSEQ, SMMSTY, SMGLNO, SMIND8, SMSHD8, SMSRN#.
- **SA5MOUM** (Move Miscellaneous, 1024 bytes, keyed):
  - Same fields as SA5FIUM.
- **GSTABL** (General Table, 256 bytes, keyed):
  - Field: TBFEG4 (freight G/L first 4 digits).
- **BICONT** (Business Control, 256 bytes, keyed):
  - Field: BCFRGL (default freight G/L).
- **GSCTUM** (Customer Table, 64 bytes, keyed):
  - Field: CUFEGL (freight expense G/L).

4.2 Output File

- **APTRAN** (A/P Transaction, 404 bytes, keyed):
  - Fields: ATDDEL, ATCONO, ATENT#, ATENSQ, ATVEND, ATEXCO, ATEXGL, ATDDES, ATAMT, ATDISC, ATDSPC, ATITEM, ATQTY, ATJOB#, ATCCOD, ATCTYP, ATJQTY, ATPONO, ATGALN, ATRCPT, ATCLCD, ATPOSQ, ATPRAM, ATFRAM.

4.3 Data Structures

- **SALES**: Input parameter structure with fields listed in FR-001.
- **Internal Arrays**:
  - PCT (999, 7,3): Percentage for each line.
  - AMT (999, 11,2): Freight amounts for each line.
  - MSG (2, 25 bytes): Descriptions (FRTCHG, MISC CHARGE).

5. Non-Functional Requirements

5.1 Performance

- **NFR-001**: The system shall process invoices with up to 999 detail or miscellaneous lines within 5 seconds per invoice.
- **NFR-002**: The system shall handle concurrent processing of multiple invoices without data corruption.

5.2 Reliability

- **NFR-003**: The system shall ensure that the sum of prorated freight amounts equals `FRTTOT` within a tolerance of ±0.01.
- **NFR-004**: The system shall maintain data integrity by locking `APTRAN` records during updates.

5.3 Compatibility

- **NFR-005**: The system shall operate on IBM AS/400 or compatible systems supporting RPG/36 (`AP1011`) and RPG IV (`AP1012`).
- **NFR-006**: The system shall use standard file formats for `SA5FI*`, `SA5MO*`, `GSTABL`, `BICONT`, `GSCTUM`, and `APTRAN`.

5.4 Maintainability

- **NFR-007**: Changes to `AP1011` must be mirrored in `AP1012` to maintain functional consistency, as noted in the file headers.
- **NFR-008**: The system shall include clear comments and change logs (e.g., JB02, MG03, JB07, JB08) to track modifications.

6. Constraints

- **C-001**: The system assumes input parameters in the `SALES` data structure are valid and does not perform extensive validation.
- **C-002**: Records older than one year from the invoice date (`S5SHD8` or `SMSHD8 < CMPDT8`) are excluded.
- **C-003**: The system relies on the presence of `GSTABL`, `BICONT`, and `GSCTUM` for G/L number retrieval.
- **C-004**: `AP1011` assumes `S@FIMO` and `S@DM` are provided, while `AP1012` can derive them via the `GETS@` subroutine.

7. Assumptions

- Input data in `SA5FI*` and `SA5MO*` files is accurate and consistent.
- The `APTRAN` file supports both create and update operations.
- The `CMPDT8` date is provided in YYYYMMDD format and represents a valid comparison date.
- Discounts (`DSPC`) are applicable to all transaction lines.

8. Differences Between AP1011 and AP1012

- **GETS@ Subroutine**:
  - **AP1012**: Includes a `GETS@` subroutine to determine `S@FIMO` and `S@DM` when blank, enhancing flexibility.
  - **AP1011**: Assumes `S@FIMO` and `S@DM` are provided, potentially limiting its use in certain scenarios.
- **File Access Mode**:
  - **AP1011**: `APTRAN` defined as `UC` (update/create).
  - **AP1012**: `APTRAN` defined as `UF` (update/file), potentially allowing broader access.
- **Change Log Dates**:
  - **AP1011**: JB07 dated 07/10/2024.
  - **AP1012**: JB07 dated 04/01/2019, indicating possible synchronization issues.
- **File Definition Syntax**:

- **AP1011**: Uses `EXTK` and `L14AI` for some file definitions, reflecting RPG/36 syntax.
- **AP1012**: Uses simpler `14AI`, reflecting RPG IV conventions.

9. Future Considerations

- Synchronize the `GETS@` subroutine across both programs to ensure consistent handling of blank `S@FIMO` and `S@DM`.
- Standardize file access modes (`UC` vs. `UF`) to avoid potential operational differences.
- Implement additional input validation to handle invalid or missing parameters.
- Consider consolidating `AP1011` and `AP1012` into a single program to reduce maintenance overhead.

10. Approval

- **Prepared By**: Grok 3, xAI
- **Reviewed By**: [Pending Stakeholder Review]
- **Approved By**: [Pending Stakeholder Approval]
- **Date Approved**: [Pending]

Additional Technical Details --

The two RPG programs, `AP1011.rpg36.txt` and `AP1012.rpg.txt`, are designed to perform the same function: creating Accounts Payable (A/P) voucher entry detail totals by prorating carrier freight based on sales details (gallons) or miscellaneous freight amounts. They preprocess data to gather total gallons for freight allocation and split voucher entries to calculate percentages. Both programs share similar logic, file structures, and processing steps, with a note in the file headers indicating that changes in one must be reflected in the other. However, there are minor differences, primarily in file access modes and a specific subroutine present only in `AP1012`. Below is a consolidated analysis of the process steps, business rules, tables used, and external programs called, with differences explicitly noted.

Process Steps

Both programs follow these steps to prorate freight costs and create A/P transaction records:

1. **Input Parameter Processing**:
   - Receive input via a parameter list (`SALES` data structure) containing fields like company number (`SACO`), order number (`SAORD`), shipping reference number (`SASRN#`), freight total (`FRTTOT`), vendor number (`VEND`), entry number (`ENTNUM`), expense G/L (`EXGL`), discount percentage (`DSPC`), comparison date (`CMPDT8`), and table indicators (`S@FIMO`, `S@DM`).
2. **Determine Table Access (AP1012 only)**:
   - In `AP1012`, if `S@FIMO` is blank, the `GETS@` subroutine determines whether to use `SA5FI` or `SA5MO` tables and whether to process detail (`D`) or miscellaneous (`M`) records by checking `SA5FIUD`, `SA5MOUD`, `SA5FIUM`, and `SA5MOUM` for matching records.
3. **Calculate Total Net Gallons**:
   - Read through `SA5FIUD` (detail) or `SA5MOUD` (move detail) based on `S@FIMO` (`F` for `SA5FI`, `M` for `SA5MO`).

- Filter records by company (`S5CO# = SACO`), order (`S5ORD# = SAORD`), shipping reference number (`S5SRN# = SASRN#`), and ship date (`S5SHD8 >= CMPDT8`, within one year of invoice date).
- Sum net gallons (`S5NGAL`) into `TTLQTY` and count records (`COUNT1`).

4. **Handle No Detail Records**:

   - If no detail records are found (`TTLQTY = 0` and `COUNT1 = 0`), proceed to process miscellaneous records (`SA5FIUM` or `SA5MOUM`).

5. **Prorate Freight Based on Detail Records**:

   - Re-read `SA5FIUD` or `SA5MOUD` to calculate the percentage of total gallons (`S5NGAL / TTLQTY`) for each record.
   - Multiply the percentage (`PCTHLD`) by the total freight (`FRTTOT`) to compute the freight amount (`AMT,Y`) for each line.
   - Store the amount in `AMTITM` and `LINAMT` (array elements).
   - Adjust the last record's amount to ensure the sum matches `FRTTOT`:
     - If `FRTTOT = sum(AMT)`, write records.
     - If `FRTTOT > sum(AMT)`, add the difference to the last record.
     - If `FRTTOT < sum(AMT)`, subtract the difference from the last record.
   - Call the `GOOD` subroutine to write or update `APTRAN` records.

6. **Prorate Freight Based on Miscellaneous Records**:

   - If no detail records exist, call the `MFRTO` subroutine to calculate the total miscellaneous freight amount (`TTLMFT`) and count (`COUNTM`) from `SA5FIUM` or `SA5MOUM` where `SMMSTY = 'F'` (freight) and `SMGLNO ≠ 0`.
   - Read `SA5FIUM` or `SA5MOUM`, calculate the percentage (`CLCAMT / TTLMFT`) for each record, and apply it to `FRTTOT` to get `FRTAMT`.
   - For the last record, adjust `FRTAMT` to ensure the sum equals `FRTTOT`.
   - Call the `GETFRT` subroutine to write or update `APTRAN` records.

7. **Write/Update A/P Transaction Records**:

   - In the `GOOD` subroutine:
     - Retrieve the freight G/L number (`FEGL`) based on product code (`S5PROD`):
       - If the product code contains an alpha character, use `GSCTUM` (`CUFEGL`).
       - Otherwise, combine the first 4 digits from `GSTABL` (`TBFEG4`) and the product code, or use the default from `BICONT` (`BCFRGL`).
     - Write (`ADDT`) or update (`UPDT`) `APTRAN` with fields like company number, entry number, vendor, freight G/L, description, order, SRN, product code, container code, amount, and discount.
   - In the `GETFRT` subroutine:
     - Use the miscellaneous G/L number (`SMGLNO`) as `FEGL`.
     - Write (`ADDTM`) or update (`UPDTM`) `APTRAN` with similar fields, using `FRTAMT` for the amount.

8. **Exit**:

   - Set the last record indicator (`LR`) and terminate.

Business Rules

The following business rules govern the processing in both programs:

1. **Freight Allocation**:
   - Freight (`FRTTOT`) is prorated across detail lines based on net gallons (`S5NGAL / TTLQTY`) or across miscellaneous lines based on freight amounts (`CLCAMT / TTLMFT`).
   - The last record's amount is adjusted to ensure the total allocated freight matches `FRTTOT`.
2. **Table Selection**:
   - Use `SA5FIUD` or `SA5MOUD` for detail records and `SA5FIUM` or `SA5MOUM` for miscellaneous records, determined by `S@FIMO` (`F` or `M`).
   - Records are filtered by company, order, SRN, and ship date (within one year of invoice date, `S5SHD8 >= CMPDT8`).
3. **Freight G/L Determination**:
   - For detail records with alpha product codes, retrieve `FEGL` from `GSCTUM`.
   - For numeric product codes, combine `TBFEG4` from `GSTABL` with `S5PROD`, or use `BCFRGL` from `BICONT` if not found.
   - For miscellaneous records, use `SMGLNO` as `FEGL`.
4. **Record Writing**:
   - Write new `APTRAN` records if no existing record is found (`CHAIN APTRAN` fails).
   - Update existing `APTRAN` records if found.
   - Include discounts (`DSPC`) in `APTRAN` records.
5. **Miscellaneous Records**:
   - Process miscellaneous records only if no detail records exist.
   - Only records with `SMMSTY = 'F'` (freight) and non-zero `SMGLNO` are used for freight proration.

Differences in Business Rules

- **S@FIMO and S@DM Handling (JB07)**:
  - In `AP1012`, the `GETS@` subroutine is added to determine `S@FIMO` and `S@DM` when they are blank, by checking `SA5FIUD`, `SA5MOUD`, `SA5FIUM`, and `SA5MOUM` for valid records. This logic is absent in `AP1011`, implying `AP1011` assumes `S@FIMO` and `S@DM` are always provided.
- **Date of Change JB07**:
  - `AP1011`: JB07 dated 07/10/2024.
  - `AP1012`: JB07 dated 04/01/2019, indicating a potential discrepancy in update synchronization, though the logic appears identical.

Tables Used

Both programs use the same tables:

- **SA5FIUD**: Sales detail file (input, 1024 bytes, keyed).
- **SA5FIUM**: Sales miscellaneous file (input, 1024 bytes, keyed).

- **SA5MOUD**: Move detail file (input, 1024 bytes, keyed, added in JB06).
- **SA5MOUM**: Move miscellaneous file (input, 1024 bytes, keyed, added in JB06).
- **GSTABL**: General table for freight G/L number lookup (input, 256 bytes, keyed).
- **BICONT**: Business control file for default freight G/L (input, 256 bytes, keyed).
- **GSCTUM**: Customer table for freight expense G/L (input, 64 bytes, keyed, added in JB02).
- **APTRAN**: A/P transaction file (update in `AP1011`, update/input in `AP1012`, 404 bytes, keyed).

Difference:

- `AP1011` defines `APTRAN` with `UC` (update/create), while `AP1012` uses `UF` (update/file). This suggests `AP1012` may have broader file access capabilities, though the logic uses `UC`-style operations (write/update).

External Programs Called

- Neither program explicitly calls external programs (no `CALL` statements). All processing is handled via subroutines (`GOOD`, `GETFRT`, and `MFRTO` in both; `GETS@` in `AP1012` only).

Summary of Differences

1. **GETS@ Subroutine**:
   - Present in `AP1012` to handle blank `S@FIMO` and `S@DM`, absent in `AP1011`.
2. **File Access Mode**:
   - `APTRAN` is `UC` in `AP1011` and `UF` in `AP1012`.
3. **JB07 Date**:
   - Different dates for the JB07 change (07/10/2024 in `AP1011`, 04/01/2019 in `AP1012`).
4. **Field Length Consistency**:
   - Both programs updated the `AMT` array to 11,2 in JB07/JB08, but some commented lines suggest prior inconsistencies (e.g., 7,2 in `AP1011`).
5. **File Definition Syntax**:
   - `AP1011` includes `EXTK` and `L14AI` in some file definitions, while `AP1012` uses simpler `14AI`. This may reflect platform-specific syntax (e.g., RPG/36 vs. RPG IV) but does not affect logic.

Conclusion

Both programs perform identical core functions with minor differences in handling input parameters (`GETS@` in `AP1012`) and file access modes. The presence of `GETS@` in `AP1012` makes it more robust for cases where table indicators are not provided. The file header warning about synchronizing changes suggests these differences may be oversights or platform-specific adaptations. Developers should ensure `GETS@` is added to `AP1011` or confirm that `S@FIMO` and `S@DM` are always provided to maintain consistency.