The `AP115.rpg36.txt` is an RPG III program designed for the IBM midrange systems (e.g., AS/400 or iSeries), called by an OCL program (e.g., `AP115.ocl36.txt`). It performs validation and editing of prepaid checks, ensuring that checks are valid, not already open (unless being voided), and that voided checks match the full check amount. The program generates a printed report (`APLIST`) listing any errors or discrepancies found during the validation process. Below is a detailed explanation of the process steps, business rules, tables used, and external programs called.

Process Steps of the RPG Program (AP115)

The `AP115` program processes prepaid check transactions by reading records from the `APCHKT` and `APCHKTX` files, validating them against `APCHKR`, and reporting errors via the `APLIST` printer file. The key steps are as follows:

1. **Initialization (Lines 0042–0054)**:
   - Executes at the detail level (`L2`) to initialize variables:
     - Retrieves the current system date and time (`TIME` instruction) and stores them in `TIMDAT`, `TIME`, and `DATE`.
     - Sets a separator (`SEP`) to `'* '` for formatting the printed report.
     - Initializes the page number (`PAGE`) to 0.
   - Validates the company number (`ATCONO`) against the `APCONT` file:
     - If found (`N98`), moves the company name (`ACNAME`) to `CONAME`.
     - If not found (`98`), sets `CONAME` to blanks.
   - At the detail level (`L1`):
     - Initializes check amount accumulator (`L1CKAM`) and void amount (`L1VOID`) to 0.
     - Clears indicators `10`, `11`, `12` (used for check status), and `81`, `91` (used for printing and error handling).

2. **Main Processing (Lines 0055–0058)**:
   - Accumulates the check amount (`ATCKAM`) into `L1CKAM` at the detail level (`L1`), setting indicators `10` (non-void check) or `11` (void check) based on the transaction type.
   - Calls the `L1TOT` subroutine to validate each check record.

3. **L1TOT Subroutine (Lines 0061–0081)**:
   - Validates the check record by chaining the check key (`ATCKEY`) to the `APCHKR` file:
     - For non-void checks (`10` indicator on):
       - If the check exists in `APCHKR` and is open (`AMCODE = 'O'`), sets indicator `91` and calls `L1PRT` to report an error ("CHECK IS ALREADY OPEN").
       - If the check does not exist (`91` on), proceeds without error.
     - For void checks (`11` indicator on):
       - If the check does not exist in `APCHKR` or is not open (`AMCODE ≠ 'O'`), sets indicator `91` and calls `L1PRT` to report an error ("CHECK MUST BE OPEN TO BE VOIDED").
       - Calculates the void amount (`L1VOID = -L1CKAM`) and compares it to the actual check amount (`AMCKAM`). If they do not match, sets indicator `12` and calls `L1PRT` to report an error ("WHOLE CHECK AMOUNT MUST BE VOIDED").
   - Ends the subroutine (`ENDL1T`).

4. **L1PRT Subroutine (Lines 0083–0099)**:
   - Prints error records to the `APLIST` printer file:
     - Sets the lower limit (`SETLL`) for `APCHKTX` using the check key (`ATKY21`).
     - Reads `APCHKTX` records in a loop (`AGNL1P` tag) until end-of-file (`09` indicator) or a key mismatch (`AXCKEY ≠ ATCKEY`).
     - For each matching record:
       - Sets indicators `80` (print detail) and `81` (control printing).
       - Writes the record to `APLIST` using the `EXCPT` operation.
       - Resets indicator `80` after printing.
     - Continues reading until all matching records are processed (`ENDL1P`).

5. **Output to APLIST (Lines 0102–0148)**:
   - Generates a formatted report with headers and detail lines:
     - **Header (L2)**:
       - Prints company name (`CONAME`), page number (`PAGE`), date (`DATE`), workstation ID (`WSID`), wire transfer description (`WIREDS`), and time (`TIME`).
       - Includes static text like "PREPAID CHECK EDIT" and column headers ("CO #", "PPD CHECK", "BANK G/L", "ENT#", "CHK AMOUNT", "ACTUAL CHECK AMOUNT").
     - **Detail Lines (80)**:
       - Prints company number (`AXCONO`), prepaid check number (`AXPPCK`), bank G/L account (`AXBKGL`), entry number (`AXENT#`), check amount (`AXCKAM`), and check date (`AXCKDT`).
     - **Total Lines (81)**:
       - Prints total check amount (`L1CKAM`) and actual check amount (`AMCKAM`).
       - Includes error messages based on indicators:
         - `10N91`: "CHECK IS ALREADY OPEN".
         - `11 91`: "CHECK MUST BE OPEN TO BE VOIDED".
         - `12 11N91`: "WHOLE CHECK AMOUNT MUST BE VOIDED".
     - Uses separator (`SEP`) for formatting between sections.

6. **Termination**:
   - The program processes all records in `APCHKT` and `APCHKTX`, generating the report and terminating when no more records are found.

Business Rules

1. **Check Validation**:
   - Non-void checks (`AMCODE ≠ 'V'`) must not already exist in `APCHKR` as open (`AMCODE = 'O'`). If they are open, an error is reported.
   - Void checks (`AMCODE = 'V'`) must exist in `APCHKR` and be open (`AMCODE = 'O'`). If not, an error is reported.
   - For void checks, the entire check amount (`L1CKAM`) must match the actual check amount (`AMCKAM`) in `APCHKR`. If not, an error is reported.

2. **Error Reporting**:

  - Errors are printed to the `APLIST` report for each invalid check, including company number, check number, bank G/L, entry number, check amount, and error message.
  - The report includes totals for check amounts and highlights discrepancies.

3. **Company Validation**:

   - The company number (`ATCONO`) must exist in `APCONT`. If not, the company name is blanked out.

4. **Formatting and Output**:

   - The report includes headers with company, date, time, and workstation details, followed by detail lines for each check and totals for check amounts.
   - Errors are clearly marked with descriptive messages to guide correction.

Tables (Files) Used

The program uses the following files, defined with specific attributes:

1. **APCHKT**:

   - Primary input file (`IP`), 80 bytes, key length 21, used to read prepaid check transactions.
   - Fields: `ATCONO` (company), `ATBKGL` (bank G/L), `ATPPCK` (prepaid check number), `ATCKAM` (check amount), `ATCKDT` (check date).

2. **APCHKTX**:

   - Indexed input file (`ID`), 80 bytes, key length 21, used to retrieve additional check details.
   - Fields: `AXCONO` (company), `AXBKGL` (bank G/L), `AXPPCK` (prepaid check number), `AXENT#` (entry number), `AXCKAM` (check amount), `AXCKDT` (check date).

3. **APCHKR**:

   - Input file (`IC`), 128 bytes, key length 16, used to validate check status.
   - Fields: `AMCODE` (check status: 'D', 'O', 'R', 'V'), `AMCKAM` (check amount).

4. **APCONT**:

   - Input file (`IC`), 256 bytes, key length 2, used to validate company number and retrieve company name.
   - Fields: `ACNAME` (company name).

5. **APLIST**:

   - Output printer file (`O`), 132 bytes, used to generate the prepaid check edit report.
   - Contains headers, detail lines, totals, and error messages.

External Programs Called

- **None**: The `AP115` program does not call any external programs. It operates independently, processing input files and generating the report.

Summary

The `AP115` RPG program validates prepaid checks by checking their status in `APCHKR` and ensuring compliance with business rules (e.g., non-void checks must not be open, void checks must be open and fully voided). It processes records from `APCHKT` and `APCHKTX`, validates against `APCONT` and `APCHKR`, and produces a detailed error report via `APLIST`. The program enforces data integrity for check processing, ensuring that only valid checks are processed and errors are clearly reported for correction. No external programs are called, making it a self-contained validation routine.