

AF010R.MBR

Path: NXCLOUD/rpgsrc/AF010R.MBR **Generated:** 2026-01-08 12:04:34 **Processing Time:** 13040ms

Business Logic for Postcode Register Maintenance

This document outlines the business rules that govern the maintenance of the postcode register, based on an analysis of the RPG program AF010R. The primary focus is on how the program manages postcode records and user interactions through a subfile interface.

The core logic for maintaining postcode records is contained within the init subroutine in AF010R. The program processes user inputs to create, update, delete, and display postcode records stored in the database.

Order Status and Header Rules

Postcode Maintenance: aposl1, aposl2, aposlr, aposlu

1. Record Creation

- Logic:** A new postcode record is created when the user inputs a new postcode and submits the form.
- File:** aposl1 (Postcode file 1)
- Field:** aposl1_ponr
- Condition:** The process will create a record if the user selects the create option (b1valg = 2).

2. Record Update

- Logic:** An existing postcode record is updated with new information when the user submits the updated data.
- File:** aposlu (Postcode update file)
- Field:** aposlu_ponr
- Condition:** The process will update a record if the user selects the update option (b1valg = 2) and the record exists.

Configuration and Authorization Rules

1. User Input Validation

- Logic:** The program checks for valid user inputs before processing any actions.
- Files:**
 - aposl1 (Postcode file 1)
 - aposl2 (Postcode file 2)
- Fields:**
 - b1ponr (Postcode number)
 - b1sted (Postcode description)
- Condition:** The program will not proceed if the postcode number is zero or if the description is blank.

2. Record Deletion

- Logic:** A postcode record is deleted when the user confirms the deletion action.

- File:** aposlu (Postcode update file)
 - Field:** aposlu_ponr
 - Condition:** The process will delete a record if the user selects the delete option (b1valg = 4).
-

Financial and Transactional Rules

1. Record Retrieval

- Logic:** The program retrieves postcode records based on user input for display in the subfile.
- File:** aposl1 (Postcode file 1)
- Fields:**
 - b1ponr (Postcode number)
 - b1sted (Postcode description)
- Condition:** The program retrieves records when the user selects the display option (b1valg = 5).

2. Subfile Management

- Logic:** The program manages the display of records in a subfile format, allowing users to navigate through records.
 - File:** b1sfl (Subfile for postcode records)
 - Condition:** The subfile is populated and displayed based on user interactions and existing records.
-

Special Conditions (Program-Specific)

1. Postcode Selection (AF010R)

- Logic:** The program allows users to select a postcode for further actions, such as viewing or editing.
- File:** aposl1 (Postcode file 1)
- Field:** aposl1_ponr
- Condition:** The selection is valid if the user has input a postcode number (b1valg = 1).

2. Error Handling (AF010R)

- Logic:** The program displays error messages for invalid operations or inputs.
 - File:** dspfbk (Display feedback structure)
 - Fields:** d_fcrn (Feedback code)
 - Condition:** Error messages are triggered based on specific validation checks throughout the program.
-

Subprogram Calls Affecting Logic

Beyond direct file checks, several external subprograms are called that play a significant role in the workflow.

1. dsp_subfile (Display Subfile)

- Trigger:** Called to display the current state of the subfile after any user action.
- Logic:** This subroutine manages the display of records in the subfile.
- Impact:** This call ensures that the user sees the most up-to-date list of postcode records.

2. clr_subfile (Clear Subfile)

- Trigger:** Called to clear the subfile before repopulating it with new data.

- Logic:** This subroutine resets the subfile's contents to prepare for new data.

- Impact:** This is a **critical step** to ensure that old data does not persist in the user interface.

3. crt_subfile (Create Subfile)

- Trigger:** Called to populate the subfile with records from the database.

- Logic:** This subroutine reads records from the postcode files and fills the subfile.

- Impact:** This represents the **main data retrieval operation** that allows users to interact with postcode records effectively.