AF010R.MBR

Path: NXCLOUD/rpgsrc/AF010R.MBR **Generated**: 2026-01-08 15:03:31 **Processing Time**: 16918ms

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 records are processed and managed within the postcode register.

The core logic for managing the postcode records is contained within the various subroutines in AF010R. The program facilitates the addition, modification, deletion, and viewing of postcode entries, utilizing subfile structures for user interaction.

Order Status and Header Rules

Postcode Maintenance: aposl1, aposl2, aposlr, aposlu, faf010d

1. Record Retrieval

•**Logic:** The program retrieves postcode records based on user input and displays them in a subfile format.

•**File:** aposl1 (Postcode file 1)

•**Field:** aposl1_ponr

•**Condition:** The program will not display a record if b1valg is set to 1, indicating that the user is retrieving a postcode.

2. Record Update

•**Logic:** The program updates existing postcode records when the user selects to modify a record.

•**File:** aposlu (Postcode update file)

•**Field:** aposlu_ponr

•**Condition:** The update occurs if b1valg is set to 2, indicating a modification request.

Configuration and Authorization Rules

1. User Authorization

•**Logic:** The program checks user permissions before allowing modifications to postcode records.

•**Files:**

•faf010d (User data structure)

•aposlr (Postcode reference file)

•**Fields:**

•l_user (Current user)

•aposlr_ponr (Postcode reference number)

•**Condition:** The program will only allow updates if the user has the necessary permissions indicated by the l_user variable.

2. Firm Association

•**Logic:** Each postcode record is associated with a specific firm, ensuring that only relevant data is accessed.

•**File:** aposl1 (Postcode file)
•**Field:** w_firm
•**Condition:** The program will filter records based on the current firm (w_firm), ensuring that only records belonging to the logged-in firm are processed.

Financial and Transactional Rules

1. Record Deletion
•**Logic:** The program allows for the deletion of postcode records when requested by the user.
•**File:** aposlu (Postcode update file)
•**Fields:**
•aposlu_ponr (Postcode number)
•b1valg (User action indicator)
•**Condition:** The deletion occurs if b1valg is set to 4, indicating a delete action.
2. Record Creation
•**Logic:** New postcode records can be created through user input via a dedicated screen.
•**File:** aposlur (Postcode creation file)
•**Condition:** The creation process is initiated when the user selects the option to add a new postcode.

Special Conditions (Program-Specific)

1. Subfile Management (AF010R)
•**Logic:** The program manages subfile operations, including displaying, clearing, and populating records.
•**File:** b1sfl (Subfile for postcode records)
•**Field:** b1ponr
•**Condition:** The subfile is refreshed and populated based on user actions and the current state of the records.
2. Error Handling (AF010R)
•**Logic:** The program includes error handling for invalid user inputs and record operations.
•**File:** dspfbk (Display feedback structure)
•**Fields:**
•d_fcrn (Current record number)
•b_feil (Error flag)
•**Condition:** If an error occurs during processing, the program will set the error flag and display an appropriate message to the user.

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 when the program needs to display the current state of the subfile.
•**Logic:** This subroutine manages the display of records in the subfile format.

•**Impact:** This call acts as a **major logical gateway**, ensuring that the user sees the most up-to-date information.

2. crt_subfile (Create Subfile)

•**Trigger:** Invoked to populate the subfile with records from the postcode database.

•**Logic:** This subroutine reads records and prepares them for display.

•**Impact:** This is a **destructive filtering step**, as it clears previous entries before loading new ones.

3. clr_subfile (Clear Subfile)

•**Trigger:** Called to clear the contents of the subfile before new data is loaded.

•**Logic:** This subroutine resets the subfile indicators and prepares for new data.

•**Impact:** This ensures that the subfile is always displaying relevant and accurate data, preventing user confusion.

This documentation provides a comprehensive overview of the business logic encapsulated within the AF010R program, detailing how postcode records are managed and the conditions under which various operations occur.