AF010R.MBR

Path: NXCLOUD/rpgsrc/AF010R.MBR **Generated**: 2026-01-08 15:42:59 **Processing Time**: 17041ms

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 the logic that handles user interactions and data management for postcode entries.

The core logic for maintaining the postcode register is contained within the various subroutines in AF010R. The program processes user commands to create, update, delete, and display postcode records, utilizing subfiles for efficient data handling.

Order Status and Header Rules

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

1. Record Selection

- **Logic:** The program allows users to select a postcode record based on user input. If the input postcode number or description is provided, it positions the record in the subfile.
- **File:** aposl1 (Postcode file)
- **Field:** aposl1_ponr
- **Condition:** The record is selected if b2ponr is not equal to 0 or b2sted is not blank.

2. Display Subfile

- **Logic:** The program displays the contents of the subfile based on the current positioning of records.
- **File:** faf010d (Workstation display file)
- **Field:** b2cmd
- **Condition:** The display is updated if w_srrn is not equal to 0.

Configuration and Authorization Rules

1. User Input Validation

- **Logic:** The program checks if the user input for postcode number is valid before proceeding with any operations.
- **Files:**
- aposlr (Postcode reference file)
- aposlu (Postcode update file)
- **Fields:**
- aposlr_ponr (from aposlr)
- aposlu_ponr (from aposlu)
- **Condition:** The program will not proceed if c1ponr is equal to 0.

2. Record Existence Check

- **Logic:** Before updating or deleting a postcode record, the program checks if the record exists in the database.

•**File:** aposlr (Postcode reference file)

•**Field:** aposlr_ponr

•**Condition:** The record must be found (%found) for the operation to proceed.

Financial and Transactional Rules

1. Record Creation

•**Logic:** When creating a new postcode record, the program inserts the new record into the database if it does not already exist.

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

•**Fields:**

•apsted (Postcode description)

•apfylk (County)

•**Condition:** A new record is created if b1valg equals 2 and the record does not already exist.

2. Record Deletion

•**Logic:** The program allows users to delete a postcode record after confirming the action.

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

•**Condition:** The record is deleted if b1valg equals 4, and the user confirms the action.

Special Conditions (Program-Specific)

1. Subfile Refresh (AF010R)

•**Logic:** The subfile is refreshed to reflect the latest data after any operation (create, update, delete).

•**File:** faf010d (Workstation display file)

•**Field:** b2cmd

•**Condition:** The subfile is cleared and repopulated whenever a record is modified.

2. Error Handling (AF010R)

•**Logic:** The program handles errors by displaying messages to the user when invalid input is detected or when operations fail.

•**File:** faf010d (Workstation display file)

•**Fields:** c1ponr (Postcode number), b_anul (Cancel flag)

•**Condition:** An error message is displayed if *inkc or *inkl is triggered, indicating the user can cancel the operation.

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 contents of the subfile after user input.

•**Logic:** This subroutine handles the display logic for the subfile based on the current record positioning.

•**Impact:** This call ensures that the user sees the most up-to-date records in the subfile.

2. crt_subfile (Create Subfile)

- **Trigger:** Invoked to populate the subfile with records from the database.
- **Logic:** This subroutine reads records from the postcode files and fills the subfile for display.
- **Impact:** It is critical for ensuring that the subfile reflects the current state of the postcode records.

3. xc2bld (Edit/Create Record)

- **Trigger:** Called when a user wants to edit or create a new postcode record.
- **Logic:** This subroutine manages the logic for validating and saving new or modified records.
- **Impact:** Represents the core functionality for maintaining the integrity of postcode records in the system.