AF010R.MBR

Path: NXCLOUD/rpgsrc/AF010R.MBR **Generated**: 2026-01-08 12:16:20 **Processing Time**: 11622ms

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 related to the handling of postcode records, including their creation, update, deletion, and display.
The core logic for maintaining postcode records is contained within the various subroutines in AF010R. The program processes records from the postcode files and manages user interactions through a subfile interface.

Order Status and Header Rules

Postcode Maintenance: aposl1, aposl2, aposlr, aposlu

1. Record Selection

•**Logic:** The program selects records based on user input for postcode or description.
•**File:** aposl1 (Postcode file 1)
•**Field:** aposl1_ponr
•**Condition:** The process will not select a record if b2ponr is not equal to 0 or b2sted is not blank.

2. Subfile Display

•**Logic:** The program displays records in a subfile format for user interaction.
•**File:** b1sfl (Subfile for postcode records)
•**Field:** b1ponr, b1sted
•**Condition:** The subfile is populated only if there are records to display.

Configuration and Authorization Rules

1. User Access Control

•**Logic:** The program checks user permissions before allowing modifications to postcode records.
•**Files:**
•aposl1 (Postcode file 1)
•aposl2 (Postcode file 2)
•**Fields:**
•w_firm (Firm identifier)
•aposl1_ponr (Postcode identifier)
•**Condition:** The user must have access rights defined by w_firm to modify records.

2. Parameter Initialization

•**Logic:** The program initializes parameters from the local data area for processing.
•**File:** aposl1 (Postcode file 1)
•**Field:** p_ponr
•**Condition:** The program requires p_ponr to be set for proper record handling.

Financial and Transactional Rules

1. Record Update Logic

•**Logic:** The program updates existing postcode records based on user input.

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

•**Fields:**

•apsted (Postcode description)

•apfylk (County)

•**Condition:** The record is updated if the user has selected an existing postcode for modification.

2. Record Deletion Logic

•**Logic:** The program deletes a selected postcode record upon user confirmation.

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

•**Condition:** The record is deleted if the user selects the delete option and confirms the action.

Special Conditions (Program-Specific)

1. Subfile Refresh Logic (AF010R)

•**Logic:** The program refreshes the subfile to reflect the latest data after any operation.

•**File:** b1sfl (Subfile for postcode records)

•**Field:** b1ponr

•**Condition:** The subfile is refreshed after any create, update, or delete operation.

2. Error Handling Logic (AF010R)

•**Logic:** The program displays error messages for invalid inputs or operations.

•**File:** c1msg (Message display file)

•**Fields:** c1ponr (Postcode)

•**Condition:** An error message is shown if the user attempts to create a postcode that already exists.

Subprogram Calls Affecting Logic

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

1. xc1bld (Subroutine for new record creation)

•**Trigger:** Called when the user initiates the creation of a new postcode record.

•**Logic:** This subroutine handles the display and validation of the new record input.

•**Impact:** This call acts as a **critical entry point** for creating new records.

2. xc2bld (Subroutine for editing records)

•**Trigger:** Called when the user selects an existing postcode for editing.

•**Logic:** This subroutine retrieves the current record details and prepares the screen for editing.

•**Impact:** This is a **key functional step** that ensures accurate updates to existing records.

3. xd1win (Subroutine for deletion confirmation)

•**Trigger:** Invoked when the user opts to delete a postcode record.

•**Logic:** This subroutine displays a confirmation window for the deletion action.

•**Impact:** This represents a **critical decision point** in the workflow, ensuring user confirmation before deletion.