AF010R.MBR

Path: NXCLOUD/rpgsrc/AF010R.MBR **Generated**: 2026-01-08 12:26:02 **Processing Time**: 12159ms

Business Logic for Post Code Register Maintenance

This document outlines the business rules that govern the maintenance of the post code register, based on an analysis of the RPG program AF010R. The primary focus is on how the program handles the display and manipulation of post code data through subfile processing.

The core logic for maintaining the post code register is contained within the various subroutines defined in AF010R. The program processes records from the post code files, allowing users to create, update, delete, and view post code entries.

Order Status and Header Rules

Post Code Maintenance: aposl1, aposl2, aposlr, aposlu

1. Record Retrieval

•**Logic:** The program retrieves records based on the user's input for post code or description.

•**File:** aposl1 (Post Code File 1)

•**Field:** aposl1_ponr

•**Condition:** The program will not retrieve a record if b2ponr is equal to 0 or b2sted is blank.

2. Record Display

•**Logic:** Displays the retrieved records in a subfile format for user interaction.

•**File:** aposl1 (Post Code File 1)

•**Field:** aposl1_ponr

•**Condition:** The subfile will be cleared and repopulated if the user navigates through the records.

Configuration and Authorization Rules

1. User Authorization

•**Logic:** The program checks if the user has the necessary permissions to perform operations.

•**Files:**

•aposl1 (Post Code File 1)

•aposlu (Post Code Update File)

•**Fields:**

•l_user (User ID from Local Data Area)

•w_firm (Firm ID)

•**Condition:** The program will only allow updates if the user ID matches the firm ID.

2. Input Validation

•**Logic:** Validates user input for creating or updating records.

•**File:** aposlu (Post Code Update File)

•**Field:** aposlu_ponr

•**Condition:** The program will raise an error if c2sted (Description) is blank during an update.

Financial and Transactional Rules

1. Record Creation

•**Logic:** Allows users to create new post code records and store them in the database.

•**File:** aposlu (Post Code Update File)

•**Fields:**

•apsted (Post Code Description)

•apfylk (County)

•**Condition:** A new record will be written if c1ponr (Post Code Number) is not already present in the database.

2. Record Deletion

•**Logic:** Deletes a selected post code record from the database.

•**File:** aposlu (Post Code Update File)

•**Condition:** The record will be deleted if the user confirms the deletion action.

Special Conditions (Program-Specific)

1. Post Code Retrieval (AF010R)

•**Logic:** Retrieves the post code based on user selection from the subfile.

•**File:** aposl1 (Post Code File 1)

•**Field:** b1ponr

•**Condition:** The program will exit if b1valg (User Selection) equals 1, indicating a retrieval action.

2. Post Code Update (AF010R)

•**Logic:** Updates an existing post code record with new values.

•**File:** aposlu (Post Code Update File)

•**Fields:**

•apsted (Post Code Description)

•apfylk (County)

•**Condition:** The update will occur if b1valg equals 2, indicating an update action.

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 records are processed.

•**Logic:** It checks if there are records to display and formats them for user interaction.

•**Impact:** This call ensures that the user sees the current state of the post code register.

2. crt_subfile (Create Subfile)

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

•**Logic:** It reads records from the post code files and prepares them for display.

•**Impact:** This is a **critical step** in ensuring that the user can interact with the latest data.

3. clr_subfile (Clear Subfile)

•**Trigger:** Called before repopulating the subfile to ensure no stale data is displayed.

- **Logic:** Clears the subfile and resets the counters for pagination.
- **Impact:** This ensures that the user interface remains responsive and accurate, avoiding confusion with old data.