

# AF010R.MBR

**Path:** NXCLOUD/rpgsrc/AF010R.MBR **Generated:** 2026-01-08 12:21:13 **Processing Time:** 12743ms

## 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 handling of postcode data through various user interactions and subfile management.

The core logic for postcode maintenance is contained within the \*inzsr subroutine in AF010R. This program interacts with multiple files to manage postcode entries, allowing for the creation, updating, deletion, and retrieval of postcode information.

### Order Status and Header Rules

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

#### 1. Postcode Retrieval

• **Logic:** The program retrieves postcode information based on user input and populates the subfile accordingly.

• **File:** aposl1 (Postcode file 1)

• **Field:** aposl1\_ponr

• **Condition:** The process will not select a record if b1valg is not equal to 1.

#### 2. Subfile Display Control

• **Logic:** The program manages the display of the subfile based on user actions and current data.

• **File:** faf010d (Workstation subfile)

• **Field:** b2cmd

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

### Configuration and Authorization Rules

#### 1. User and Firm Identification

• **Logic:** The program initializes user and firm identifiers for processing.

• **Files:**

• aposl1 (Postcode file 1)

• aposl2 (Postcode file 2)

• **Fields:**

• l\_user (User identifier)

• l\_firm (Firm identifier)

• **Condition:** The program retrieves values from the local data area to set user and firm identifiers at the start.

#### 2. Input Validation for New Records

• **Logic:** The program checks for valid input before allowing the creation of new postcode entries.

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

- **Field:** aposlr\_ponr
  - **Condition:** The program will prompt an error if the postcode number (c1ponr) is blank during the creation process.
- 

## Financial and Transactional Rules

### 1. Postcode Update Logic

- **Logic:** The program updates existing postcode records based on user input and validation.
- **File:** aposlu (Postcode update file)
- **Fields:**
  - apsted (Postcode description)
  - apfylk (County)
- **Condition:** The update will only occur if the postcode exists in the database.

### 2. Postcode Deletion Logic

- **Logic:** The program allows for the deletion of postcode records upon user confirmation.
  - **File:** aposlr (Postcode reference file)
  - **Condition:** The record will be deleted if the user selects the delete option (b1valg = 4).
- 

## Special Conditions (Program-Specific)

### 1. Subfile Management (AF010R)

- **Logic:** The program manages the subfile for displaying postcode records, including pagination and navigation.
- **File:** faf010d (Workstation subfile)
- **Field:** w\_srrn
- **Condition:** The subfile will be cleared and filled based on the current record position and user actions.

### 2. Postcode Copy Functionality (AF010R)

- **Logic:** The program provides functionality to copy existing postcode records to create new entries.
  - **File:** aposlu (Postcode update file)
  - **Fields:** k1ponr (Key for copying), s\_ponr (Source postcode)
  - **Condition:** The copy operation will only proceed if the source postcode exists in the database.
- 

## 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 during the display of the subfile records.
- **Logic:** This subroutine handles the display logic for the subfile, ensuring proper pagination and record visibility.
- **Impact:** This call acts as a **major logical gateway** for user interaction with the postcode records.

### 2. crt\_subfile (Create Subfile)

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

- Logic:** This subroutine reads from the postcode files and fills the subfile based on current user selections.

- Impact:** This is a **critical step** in ensuring that users see the correct data in the subfile.

### **3. xc2bld (Maintenance Program Call)**

- Trigger:** Called when a user wants to create or update a postcode entry.

- Logic:** This subroutine manages the logic for creating or updating records in the postcode database.

- Impact:** Represents the handoff to the next major business function, ensuring data integrity and proper updates.