

# AF010R.MBR

**Path:** NXCLOUD/rpgsrc/AF010R.MBR **Generated:** 2026-01-08 11:56:15 **Processing Time:** 12991ms

## 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 the program handles the display and manipulation of postcode records through a subfile interface.

The core logic for maintaining postcode records is contained within the init subroutine in AF010R. The program utilizes subfiles to display records and allows for various operations such as creating, updating, deleting, and copying postcode entries.

### Order Status and Header Rules

Postcode Register: aposl1, aposl2, aposlr, aposlu

#### 1. Record Selection

- **Logic:** The program positions the database based on the postcode or description entered by the user and populates the subfile accordingly.
- **Files:** aposl1 (Postcode table), aposl2 (Description table), aposlr (Read table), aposlu (Update table)
- **Field:** aposl1\_ponr, aposl2\_sted
- **Condition:** The subfile is populated if either b2ponr is not equal to 0 or b2sted is not blank.

#### 2. Record Retrieval

- **Logic:** The program retrieves the postcode based on user input and displays the relevant information.
- **File:** aposlr (Read table)
- **Field:** aposlr\_ponr
- **Condition:** The record is fetched if the user selects a postcode entry.

### Configuration and Authorization Rules

#### 1. User and Firm Identification

- **Logic:** The program initializes user and firm identifiers from the local data area.

##### • **Files:**

- l\_user (User identifier)
- l\_firm (Firm identifier)

##### • **Fields:**

- l\_user
- l\_firm

- **Condition:** These identifiers are set at the start of the program to ensure proper context for database operations.

#### 2. Subfile Control Initialization

- **Logic:** The program initializes the subfile control indicators to manage the display and interaction with the subfile.

- **File:** b2ctl (Control record for the subfile)
- **Field:** \*in14, \*in15
- **Condition:** The control indicators are set to manage the display of subfile records.

## Financial and Transactional Rules

### 1. Record Update

- **Logic:** The program updates postcode records based on user input and validates the changes before committing them to the database.
- **File:** aposlu (Update table)
- **Fields:**
  - apsted (Postcode description)
  - apfylk (County)
- **Condition:** The record is updated if the user selects to modify an existing postcode entry.

### 2. Record Deletion

- **Logic:** The program allows users to delete postcode records after confirming the action.
- **File:** aposlu (Update table)
- **Condition:** The record is deleted if the user selects the delete option.

## Special Conditions (Program-Specific)

### 1. Subfile Refresh (AF010R)

- **Logic:** The program refreshes the subfile display after any operation (create, update, delete) to ensure the user sees the latest data.
- **File:** b1sfl (Subfile for postcode records)
- **Field:** b1ponr, b1sted
- **Condition:** The subfile is refreshed whenever an operation is performed on the records.

### 2. Error Handling (AF010R)

- **Logic:** The program handles errors during operations by displaying appropriate messages to the user.
- **File:** c1msg (Message display)
- **Fields:** c1ponr (Postcode number)
- **Condition:** Error messages are displayed 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 (Maintenance Program for New Row)

- **Trigger:** Called when the user selects to create a new postcode entry.
- **Logic:** This subroutine handles the input and validation of new postcode data.
- **Impact:** This call ensures that new entries are validated before being added to the database.

### 2. xc2bld (Maintenance Program for Update/View)

- **Trigger:** Called when the user selects to update or view an existing postcode entry.

- Logic:** This subroutine retrieves existing data for the selected postcode and prepares it for editing.

- Impact:** This is a critical step for ensuring that users can modify existing records accurately.

### **3. xd1win (Delete Confirmation Window)**

- Trigger:** Called when the user selects to delete a postcode entry.

- Logic:** This subroutine displays a confirmation window to ensure the user intends to delete the record.

- Impact:** This represents a safeguard against accidental deletions, enhancing data integrity.