

# RY820R.MBR

**Path:** NXCLOUD/rpgsrc/RY820R.MBR **Generated:** 2026-01-08 12:59:02 **Processing Time:** 8804ms

## Business Logic for RY820R

This document outlines the business rules that govern the updating of the match register, based on an analysis of the RPG program RY820R. The primary focus is on how the program processes records from the match register and updates them with information from the item register.

The core logic for updating the match register is contained within the main program logic in RY820R. The program reads through the match register and updates items with information from the item register.

## Match Register and Item Update Rules

**RY820R:** fjmatl1, fjvarl1, fjmatlu

### 1. Check for Blank Item Code

- Logic:** The program skips the current item if the item code (jovar2) is blank.
- File:** fjmatl1 (Match Register)
- Field:** jovar2
- Condition:** The process will not select a record if jovar2 is equal to \*blank.

### 2. Update Item Information

- Logic:** If the item exists in the item register and the old item code (joldo2) does not match the new item code (jvldor), the program updates the match register with the new item information.
- File:** fjvarl1 (Item Register)
- Field:** jvarl1\_vare
- Condition:** The update occurs if the item is found (\*in91 = \*off) and the old item code is different from the new item code.

## Initialization and Key Setup Rules

### 1. Initialize Keys for Match Register

- Logic:** The program initializes the keys used for reading the match register, item register, and updating the match register.
- Files:**
  - fjmatl1 (Match Register)
  - fjvarl1 (Item Register)
  - fjmatlu (Match Update Register)
- Fields:**
  - w\_firm (from fjmatlu)
  - jvarl1\_vare (from fjvarl1)
- Condition:** Keys are set up at the beginning of the program to facilitate record access.

### 2. Retrieve Firm Information

- Logic:** The program retrieves the firm information from the local data area (LDA) for use in key operations.

- File:** N/A (Local Data Area)
- Field:** l\_firm
- Condition:** The firm information is assigned to the variable w\_firm at the start of the program.

## Special Conditions (Program-Specific)

### 1. End of Program Logic

- Logic:** The program sets the last record indicator (\*inlr) to on and returns control to the calling program.
- File:** N/A
- Condition:** This is executed at the end of the program to ensure proper termination.

### 2. Subroutine for Program Initialization

- Logic:** This subroutine initializes keys for accessing the match register and item register, as well as retrieving firm information.
- File:** N/A
- Fields:**
  - w\_firm (Firm information)
  - jvarl1\_vare (Item code)
- Condition:** This subroutine is called at the beginning of the program to prepare for processing.

## Subprogram Calls Affecting Logic

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

### 1. N/A (No external subprograms identified)

- Trigger:** N/A
- Logic:** N/A
- Impact:** N/A

### 2. N/A (No external subprograms identified)

- Trigger:** N/A
- Logic:** N/A
- Impact:** N/A

### 3. N/A (No external subprograms identified)

- Trigger:** N/A
- Logic:** N/A
- Impact:** N/A

This program does not appear to call any external subprograms, focusing solely on its internal logic for processing records.