

# RY820R.MBR

**Path:** NXCLOUD/rpgsrc/RY820R.MBR **Generated:** 2026-01-08 13:33:01 **Processing Time:** 9792ms

## Business Logic for RY820R

This document outlines the business rules that govern the process of updating the match register based on an analysis of the RPG program RY820R. The primary focus is on how items are updated in the match register using information from the item register.

The core logic for updating the match register is contained within the main program body of 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 checks if the item code (jovar2) is blank. If it is, the program skips to the next item.
- File:** fjmatl1 (Match Register)
- Field:** jovar2
- Condition:** The process will not update a record if jovar2 is blank.

### 2. Update Match Register with Item Information

- Logic:** If the item code is not blank, the program retrieves the corresponding item information from the item register (fjvarl1). If the item exists and the old value (joldo2) differs from the new value (jvldor), it updates the match register with the new item value (jmatlu\_var1).
- File:** fjvarl1 (Item Register)
- Field:** jvarl1\_vare
- Condition:** The update occurs if the item exists and the old value is different from the new value.

## Key Initialization Rules

### 1. Initialize Keys for File Access

- Logic:** The program initializes keys for accessing the match register and item register. It sets the keys based on the firm identifier and the item code.
- Files:**
  - fjmatl1 (Match Register)
  - fjvarl1 (Item Register)
- Fields:**
  - w\_firm (Firm Identifier from Local Data Area)
  - jvarl1\_vare (Item Code from Item Register)
- Condition:** Keys are initialized at the beginning of the program to facilitate record access.

### 2. Retrieve Firm Identifier

- Logic:** The program retrieves the firm identifier from the local data area (l\_firm) and assigns it to the variable w\_firm.

- File:** Local Data Area
- Field:** l\_firm
- Condition:** This step is essential for ensuring that updates are performed for the correct firm.

## Program Flow and Control Logic

### 1. Main Loop for Processing Items

- Logic:** The program enters a loop where it reads each record in the match register. It continues processing until there are no more records to read.
- File:** fpmat1 (Match Register)
- Condition:** The loop continues as long as the read operation does not encounter the end of the file.

### 2. End of Program Handling

- Logic:** Upon completion of processing, the program sets the last record indicator (\*inlr) to on and returns control to the calling program.
- File:** None
- Condition:** This step ensures proper termination of the program.

## Subprogram Calls Affecting Logic

Beyond direct file checks, several key initialization steps are performed that play a significant role in the workflow.

### 1. Initialization Subroutine (\*inzsr)

- Trigger:** This subroutine is called at the beginning of the program.
- Logic:** It initializes keys for accessing the match register and item register, and retrieves the firm identifier from the local data area.
- Impact:** This initialization is crucial for ensuring that the program can correctly access and update records in the match register based on the item register.

This documentation provides a clear overview of the business logic implemented in the RY820R program, outlining the rules and conditions that govern the updating of the match register with item information.