

RY820R.MBR

Path: NXCLOUD/rpgsrc/RY820R.MBR **Generated:** 2026-01-09 10:13:45 **Processing Time:** 10846ms

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 the program reads through the match register and updates items 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 processes records from the match register and updates them with relevant information from the item register.

Order Status and Header 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 select a record if jovar2 is equal to *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) and updates the match register (fjmatlu) if the conditions are met.
- **File:** fjvarl1 (Item Register)
- **Field:** jvarl1_vare
- **Condition:** The update occurs if the item is found in fjvarl1 and the old value (joldo2) is different from the new value (jvldor).

Configuration and Authorization Rules

1. Key Initialization for Match Register

- **Logic:** The program initializes keys for reading the match register, looking up items, and updating the match register.

• **Files:**

- fjmatl1 (Match Register)

- fjvarl1 (Item Register)

• **Fields:**

- w_firm (from LDA)

- jvarl1_vare (from fjvarl1)

- **Condition:** Keys are initialized at the start of the program to facilitate record access.

2. Firm Information Retrieval

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

- **File:** N/A
- **Field:** l_firm
- **Condition:** This retrieval occurs at the beginning of the program to set the context for processing.

Financial and Transactional Rules

1. Conditional Update of Match Register

• **Logic:** The program updates the match register with new item information only if the item exists in the item register and the old value differs from the new value.

• **File:** fmatlu (Updated Match Register)

• **Fields:**

• jmatlu_var1 (New Item Code)

• joldo2 (Old Item Code)

• **Condition:** The update is executed if the item is found and the old item code is different from the new item code.

2. End of Program Logic

• **Logic:** The program sets the last record indicator to on and returns control to the calling program.

• **File:** N/A

• **Condition:** This occurs at the end of the processing loop.

Special Conditions (Program-Specific)

1. Initialization Subroutine (RY820R)

• **Logic:** The initialization subroutine sets up keys for accessing the match register and item register and retrieves firm information from the local data area.

• **File:** N/A

• **Field:** w_firm

• **Condition:** This initialization is necessary before processing records.

2. Record Processing Loop (RY820R)

• **Logic:** The program enters a loop to read records from the match register and processes each record according to the defined logic.

• **File:** fmatl1 (Match Register)

• **Fields:** jmatl1_key, jvarl1_vare

• **Condition:** The loop continues until all records have been processed.

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 called)

• **Trigger:** N/A

• **Logic:** N/A

• **Impact:** The program operates independently without external subprogram dependencies.

This documentation provides a clear overview of the business logic implemented in the RY820R program, detailing the rules and conditions governing the update process for the match register.