

RY820R.MBR

Path: NXCLOUD/rpgsrc/RY820R.MBR **Generated:** 2026-01-08 12:35:24 **Processing Time:** 11185ms

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 and updates items in the match register using information from the item register.

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

Order Status and Header Rules

RY820R: fjmatl1, fjvarl1, fjmatlu

1. Skip Blank Items

- **Logic:** The program skips any items in the match register where the item description (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 Match Register Information

- **Logic:** If the item exists in the item register and the old item description is different from the new one, 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 in fjvarl1 and joldo2 is not equal to 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 set based on the firm and item identifiers.

2. Firm Information Retrieval

- **Logic:** The program retrieves the firm information from the local data area (LDA).
- **File:** N/A
- **Field:** l_firm

- **Condition:** This is executed at the beginning of the program to set the firm context.

Financial and Transactional Rules

1. Chain to Item Register

- **Logic:** The program chains to the item register to retrieve item details based on the item identifier.

- **File:** fjvarl1 (Item Register)

- **Fields:**

- jvarl1_vare (Item Identifier)

- joldo2 (Old Item Description)

- **Condition:** The chain is successful if the item exists in the register.

2. Update Match Register Entry

- **Logic:** If the item is found and the old description differs from the new one, the match register entry is updated with the new description.

- **File:** fmatlu (Match Update)

- **Condition:** The update occurs if the chain to fmatlu is successful.

Special Conditions (Program-Specific)

1. End of Program Handling (RY820R)

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

- **File:** N/A

- **Field:** N/A

- **Condition:** This is executed at the end of the program to ensure proper termination.

2. Initialization Subroutine (RY820R)

- **Logic:** The subroutine initializes keys for the match register and retrieves the firm information from the local data area.

- **File:** N/A

- **Fields:** w_firm (Firm Identifier), jvarl1_vare (Item Identifier)

- **Condition:** This is executed at the start 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 processing within its own logic.