VL001R.MBR

Path: NXCLOUD/rpgsrc/VL001R.MBR **Generated**: 2026-01-09 10:13:45 **Processing Time**: 11355ms

Business Logic for Inventory Update Process

This document outlines the business rules that govern the inventory update process, based on an analysis of the RPG program VL001R. The primary focus is on how inventory records are updated and maintained, including historical inventory records.

The core logic for updating inventory is contained within the *inzsr subroutine in VL001R. The program processes transactions related to inventory adjustments, sales, and stock movements, ensuring that the inventory records are accurately maintained.

Inventory Status and Validation Rules

Inventory Update Process: vlaglu, vhislu, vvarl1, vugrl1, vltyl1, votyl1

1. Valid Inventory Check

- **Logic:** The program checks if the inventory item exists in the inventory register. If not, the process is aborted.
- **File:** vvarl1 (Inventory Item Register)
- **Field:** vvarl1_vare
- **Condition:** The process will not proceed if the inventory item is not found (*in90 is on).

2. Valid Warehouse Check

- **Logic:** The program verifies if the specified warehouse is valid. If the warehouse is invalid, a log entry is created, and the process is aborted.
- **File:** ra10l1 (Warehouse Register)
- **Field:** ra10l1_jkod
- **Condition:** The process will not proceed if the warehouse is not found (*in90 is on).

Transaction Type and Processing Rules

1. Transaction Type Validation

- **Logic:** The program checks if the transaction type and alternative processing code are valid. If both are blank, the process is aborted.
- **File:** Not applicable
- **Field:** hlotyp, hlaltk
- **Condition:** The process will not proceed if both fields are blank.

2. Delivery Type Check

- **Logic:** If the delivery type is '1', the program will not update the inventory.
- **File:** Not applicable
- **Field:** hllety
- **Condition:** The process will not proceed if hllety equals '1'.

Inventory Adjustment and Update Rules

1. Inventory Adjustment Logic

- **Logic:** Depending on the transaction type, the program updates the inventory levels accordingly. Adjustments are made based on the transaction type and the sign of the quantity.
- **File:** vlaglu (Inventory Update Register)
- **Fields:**
- hlanta (Quantity)
- vlbehl (Current Inventory Level)
- **Condition:** The inventory is updated based on the rules defined for each transaction type.

2. Historical Record Creation

- **Logic:** The program creates a historical record of the inventory transaction for tracking purposes.
- **File:** vhislu (Historical Inventory Register)
- **Condition:** A historical record is created for every inventory adjustment made.

Special Conditions (Program-Specific)

1. MalProff Update Logic (VL001R)

- **Logic:** The program updates additional records for MalProff if the transaction type indicates a MalProff update.
- **File:** rmpil1, rmpilu (MalProff Additional Records)
- **Condition:** If the transaction type is 'MP', additional updates are performed.

2. Inventory Unit Conversion Logic (VL001R)

- **Logic:** The program converts quantities based on the relationship between sold units and inventory units.
- **File:** vvenl1 (Inventory Unit Register)
- **Condition:** The conversion is performed if the sold unit differs from the inventory unit.

Subprogram Calls Affecting Logic

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

1. CO402R (MalProff Update Program)

- **Trigger:** Called during the initialization phase of the program.
- **Logic:** This program checks if the MalProff update is required based on the transaction type.
- **Impact:** If the update is required, it opens the necessary files for further processing.

2. VL710R (Inventory Type Retrieval Program)

- **Trigger:** Called to retrieve the inventory type based on the item and warehouse.
- **Logic:** This program fetches the inventory type and related data for processing.
- **Impact:** It ensures that the correct inventory type is used for the transaction.

3. oppr_hist (Historical Record Creation Subroutine)

- **Trigger:** Called to create a historical record after an inventory update.
- **Logic:** This subroutine writes the transaction details to the historical inventory register.
- **Impact:** It maintains a comprehensive history of inventory transactions for auditing and tracking purposes.

This documentation provides a detailed overview of the business logic implemented in the VL001R program, focusing on inventory updates and their associated rules and conditions.