RV001R.MBR

Path: NXCLOUD/rpgsrc/RV001R.MBR **Generated**: 2026-01-08 13:32:58 **Processing Time**: 11820ms

Business Logic for RV001R

This document outlines the business rules that govern the processing of invoice lines from the FOVF file into a comma-separated format, based on an analysis of the RPG program RV001R. The primary focus is on the data transformation and validation logic involved in this process.

The core logic for reading and processing invoice lines is contained within the control_init subroutine in RV001R. The program reads records from various input files, performs validations, and prepares the data for output.

Order Status and Header Rules

RV001R: FOVF, SOHEL1, VOTYL1, AFORL1, RB10L1, RkunL1, RB00L1

1. Firm Check

- **Logic:** The process will skip processing if the firm identifier does not match the expected value.
- **File:** FOVF (Invoice File)
- **Field:** l_firm
- **Condition:** "The process will not select a record if l_firm is not equal to fffirm."

2. Blank Field Initialization

- **Logic:** Initializes various fields to blank if the firm check is passed.
- **File:** FOVF (Invoice File)
- **Field:** Various fields including d_dati, w_lnavn, w_lnav, etc.
- **Condition:** "Fields are set to blank after confirming the firm matches."

Configuration and Authorization Rules

1. Tax Code Assignment

- **Logic:** Assigns a tax code based on the availability of a specific field.
- **Files:**
- FOVF (Invoice File)
- RB00L1 (Tax Code File)
- **Fields:**
- ffdmom (from FOVF)
- w_mkod (from RB00L1)
- **Condition:** "If ffdmom is blank, w_mkod is assigned from ffcmom; otherwise, it uses ffdmom."

2. Invoice Type Conversion

- **Logic:** Checks if the invoice type needs to be converted based on the mapping in the RB10L1 file.
- **File:** RB10L1 (Invoice Type Mapping)
- **Field:** ffbilk
- **Condition:** "The process checks if the invoice type exists in the RB10L1 file and assigns it accordingly."

Financial and Transactional Rules

1. Invoice Date Handling

•**Logic:** Moves the invoice date from the input structure to the output structure.

•**File:** FOVF (Invoice File)

•**Fields:**

•ffbdat (Invoice Date)

•d_dati (Output Date Structure)

•**Condition:** "The invoice date is directly transferred to the output structure."

2. Line Item Amount Calculation

•**Logic:** Calculates the total amount for the invoice line items.

•**File:** FOVF (Invoice File)

•**Condition:** "The total amount is calculated based on the line item details."

Special Conditions (Program-Specific)

1. Delivery Address Handling (RV001R)

•**Logic:** Processes the delivery address and checks for commas in the name field.

•**File:** RkunL1 (Customer File)

•**Field:** rknavn

•**Condition:** "If the delivery name contains a comma, it splits the name into two parts."

2. Invoice Number Handling (RV001R)

•**Logic:** Handles the invoice number and ensures it is formatted correctly.

•**File:** FOVF (Invoice File)

•**Fields:** ffbiln (Invoice Number), w_ponr (Output Invoice Number)

•**Condition:** "The invoice number is formatted and assigned to the output structure."

Subprogram Calls Affecting Logic

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

1. AK710R (Invoice Processing Subroutine)

•**Trigger:** Called during the processing of invoice lines.

•**Logic:** This subroutine retrieves additional data based on the invoice and customer identifiers.

•**Impact:** "This call acts as a **data enrichment step** for invoice processing."

2. control_init (Initialization Subroutine)

•**Trigger:** Called at the beginning of the processing to initialize variables and set keys.

•**Logic:** Initializes various fields and sets up keys for file access.

•**Impact:** "This is a **critical setup step** that prepares the program for processing."

3. chain Operations (File Access)

•**Trigger:** Used throughout the program to access records in various files.

•**Logic:** Retrieves records based on defined keys.

•**Impact:** "These operations are essential for fetching related data needed for processing."

This documentation provides a comprehensive overview of the business logic implemented in the RV001R program, ensuring clarity on how data is processed and transformed within the system.