

RV001R.MBR

Path: NXKORR/rpgsrc/RV001R.MBR **Generated:** 2026-01-08 15:53:46 **Processing Time:** 19992ms

Business Logic for RV001R

This document outlines the business rules that govern the processing of invoice data from the file FOFV into a comma-separated format, based on an analysis of the RPG program RV001R. The primary focus is on the transformation and validation of invoice records as they are read and processed.

The core logic for reading and transforming invoice data is contained within the control_init subroutine in RV001R. The program reads records from the FOFV file, processes them according to various business rules, and outputs the results in a specified format.

Order Status and Header Rules

RV001R: FOFV, SOHEL1, VOTYL1, AFORL1, RB10L1, RKUNL1, RB00L1

1. Firm Check

• **Logic:** The program checks if the firm identifier from the input record matches the expected firm identifier. If they do not match, the processing for that record is skipped.

• **File:** FOFV (Invoice Data)

• **Field:** l_firm

• **Condition:** "The process will not select a record if l_firm is not equal to fffirm."

2. Clear Blank Fields

• **Logic:** If the firm matches, several fields are initialized to blank values to prepare for data processing.

• **File:** FOFV (Invoice Data)

• **Field:** Various fields (e.g., d_dati, w_lnavn, w_ladr1, etc.)

• **Condition:** "Fields are cleared if the firm matches."

Configuration and Authorization Rules

1. Tax Code Assignment

• **Logic:** The program assigns a tax code based on the input record. If the tax code field is blank, it uses a default tax code.

• **Files:**

• FOFV (Invoice Data)

• RB00L1 (Tax Code Mapping)

• **Fields:**

• ffdmom (from FOFV)

• w_mkod (to be assigned)

• **Condition:** "If ffdmom is blank, assign ffcmom to w_mkod; otherwise, assign ffdmom."

2. Invoice Type Conversion

• **Logic:** The program checks if the invoice type needs to be converted based on a mapping in the RB10L1 file.

- **File:** RB10L1 (Invoice Type Mapping)
- **Field:** ffbilk
- **Condition:** "The invoice type is checked against the mapping; if found, it is converted."

Financial and Transactional Rules

1. Invoice Date Handling

- **Logic:** The program processes the invoice date and assigns it to the appropriate fields for further processing.
- **File:** FOVF (Invoice Data)
- **Fields:**
 - ffbdat (Invoice Date)
 - d_dati (Processed Date)
- **Condition:** "The invoice date is moved to d_dati for further processing."

2. Amount Processing

- **Logic:** The program sums the amounts from the invoice for reporting purposes.
- **File:** FOVF (Invoice Data)
- **Condition:** "The amount is added to the total for the invoice."

Special Conditions (Program-Specific)

1. Mail Address Handling (RV001R)

- **Logic:** The program checks if a mail address is provided for the customer and processes it accordingly.
- **File:** RKUNL1 (Customer Data)
- **Field:** rkemal
- **Condition:** "If a mail address exists, it is included in the output."

2. Customer Name Processing (RV001R)

- **Logic:** The program checks for commas in the customer name and splits it into first and last names if necessary.
- **File:** RKUNL1 (Customer Data)
- **Fields:**
 - rknnavn (Full Customer Name)
 - wknnavu (Processed Customer Name)
- **Condition:** "If a comma is found in rknnavn, it splits the name into w2navn and w2fnav."

Subprogram Calls Affecting Logic

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

1. AK710R (Customer Data Retrieval)

- **Trigger:** Called when specific customer data needs to be retrieved based on the invoice details.
- **Logic:** This program retrieves additional customer information based on the customer ID.
- **Impact:** "This call enriches the invoice data with customer details."

2. control_init (Initialization Subroutine)

- **Trigger:** Called at the beginning of the processing to initialize variables and set up keys for file access.
- **Logic:** Initializes key fields and prepares for reading invoice header data.
- **Impact:** "This is a critical step for ensuring that all necessary data is correctly initialized before processing."

3. sohel1_key (Invoice Header Lookup)

- **Trigger:** Used to position the file pointer for reading the invoice header.
- **Logic:** Sets the key fields for the invoice header lookup.
- **Impact:** "This establishes the context for reading invoice details, ensuring accurate data retrieval."