RF010R.MBR

Path: NXCLOUD/rpgsrc/RF010R.MBR **Generated**: 2026-01-08 12:35:13 **Processing Time**: 15018ms

Business Logic for Bilagsregistrering

This document outlines the business rules that govern the process of registering vouchers (bilagsregistrering), based on an analysis of the RPG program RF010R. The primary focus is on the validation and processing of financial transactions related to vouchers.

The core logic for voucher registration is contained within the a2taga subroutine in RF010R. This subroutine handles the initialization of the registration process, including the retrieval of necessary data and the validation of input fields.

Order Status and Header Rules

Bilagsregistrering: RBUNL1, RTRAL1, RAA1L1, RLEVL1, RKUML1

1. Check for Existing Voucher Number

- **Logic:** The process will not allow a voucher number to be reused within the same year for both customer and supplier postings.
- **File:** RTRAL1 (Transaction Records)
- **Field:** RTBILN (Voucher Number)
- **Condition:** The process will block registration if the voucher number already exists for the current year.

2. Validate Voucher Date

- **Logic:** The voucher date must be valid and cannot be in the future or more than one year in the past.
- **File:** RTRAL1 (Transaction Records)
- **Field:** RTBDAT (Voucher Date)
- **Condition:** The process will not accept a voucher date that is greater than the current date or less than one year prior.

Configuration and Authorization Rules

1. Authorization for Posting

- **Logic:** Only authorized users can register vouchers; specific checks are made against user roles.
- **Files:**
- RAA1L1 (User Authorization)
- RAA3L1 (User Roles)
- **Fields:**
- L_USER (Current User)
- RAA1L1_AKOD (Authorization Code)
- **Condition:** The process will block registration if the user does not have the necessary permissions.

2. Check for Valid Account Types

- **Logic:** The account types for debit and credit must be valid and cannot both be receivable accounts.

- **File:** RLEVL1 (Supplier Records)
- **Field:** RLEVL1_LEVR (Supplier Account)
- **Condition:** The process will not allow both accounts to be receivable accounts.

Financial and Transactional Rules

1. Negative Amount Check

- **Logic:** The voucher amount cannot be negative.
- **File:** RTRAL1 (Transaction Records)
- **Fields:**
- RTBELØ (Amount)
- **Condition:** The process will block registration if the amount is less than zero.

2. Currency Validation

- **Logic:** If a currency code is provided, it must exist in the currency register.
- **File:** RA16L1 (Currency Codes)
- **Condition:** The process will not accept a currency code that is not found in the register.

Special Conditions (Program-Specific)

1. Handling of Bunt Number

- **Logic:** The program allows for the registration of a new batch (bunt) number or the retrieval of an existing one.
- **File:** RBUNL1 (Batch Records)
- **Field:** RBUNL1_BUNT (Batch Number)
- **Condition:** The process will prompt the user if a batch number is not provided.

2. Validation of KID Number

- **Logic:** The KID number must be validated using a modulus check.
- **File:** RTRA1 (KID Records)
- **Fields:** KID (KID Number)
- **Condition:** The process will block registration if the KID number fails the modulus check.

Subprogram Calls Affecting Logic

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

1. AA002R (Firm Selection Program)

- **Trigger:** Called during the initialization phase to select the firm.
- **Logic:** This program retrieves and sets the current firm based on user input.
- **Impact:** This call ensures that the correct firm context is used for all subsequent operations.

2. RF011C (Label Printing Program)

- **Trigger:** Called when a transaction is registered.
- **Logic:** This program generates labels for transactions that require printing.
- **Impact:** This ensures that all registered transactions can be properly documented.

3. RS701R (Voucher Code Validation)

- **Trigger:** Called to validate the voucher code entered by the user.

- **Logic:** This program checks if the voucher code is valid against the code register.
- **Impact:** This is a critical validation step that prevents the registration of invalid vouchers.

Use clear, business-friendly language. Fill in all sections with details from the code. Output only the Markdown.