RF010R.MBR

Path: NXCLOUD/rpgsrc/RF010R.MBR **Generated**: 2026-01-08 15:04:16 **Processing Time**: 19149ms

Business Logic for Bilagsregistrering

This document outlines the business rules that govern the process of registering financial documents (bilagsregistrering), based on an analysis of the RPG program RF010R. The primary focus is on the validation and processing of financial transactions, including checks for account types, transaction dates, and amounts.

The core logic for bilagsregistrering is contained within the a2taga subroutine in RF010R. This subroutine handles the initialization and display of the main registration screen, processes user input, and validates the data against various business rules.

Order Status and Header Rules

Bilagsregistrering: RBUNL1, RTRAAL, RAA1L1, RLEVL1, RKTRL3

1. Bunt Number Validation

- **Logic:** The process will not select a record if the bunt number is zero or if the year and period are not specified correctly.
- **File:** RBUNL1 (Bunt Header)
- **Field:** RBUNL1_BUNT
- **Condition:** "The process will not select a record if RBUNL1_BUNT is equal to *ZERO."

2. Year and Period Validation

- **Logic:** The year must be four digits and the period must be between 1 and 13.
- **File:** RTRAAL (Transaction Records)
- **Field:** A2RAAR, A2RPER
- **Condition:** "The process will not allow a record if A2RAAR is less than 2000 or A2RPER is not between 1 and 13."

Configuration and Authorization Rules

1. Account Type Validation

- **Logic:** Both debit and credit accounts cannot be receivable accounts simultaneously.
- **Files:**
- RLEVL1 (Supplier Accounts)
- RKTRL3 (Customer Accounts)
- **Fields:**
- A5DKTO (Debit Account)
- A5CKTO (Credit Account)
- **Condition:** "The process will not allow both A5DKTO and A5CKTO to be receivable accounts."

2. Transaction Date Validation

- **Logic:** The transaction date must not exceed the current year or be earlier than the previous year.
- **File:** RTRAAL (Transaction Records)

- **Field:** A5BDAT
- **Condition:** "The process will not allow a record if A5BDAT is less than the current date or greater than next year."

Financial and Transactional Rules

1. Amount Validation

- **Logic:** The amount cannot be negative, and either the amount or the currency amount must be specified.
- **File:** RTRAAL (Transaction Records)
- **Fields:**
- A5BELØ (Amount)
- A5VALB (Currency Amount)
- **Condition:** "The process will not allow a record if both A5BELØ and A5VALB are zero."

2. Duplicate Bilag Number Check

- **Logic:** The bilag number must be unique within the specified year.
- **File:** RKTRL3 (Customer Transactions)
- **Condition:** "The process will not allow a record if the bilag number has been used in the current year."

Special Conditions (Program-Specific)

1. Currency Code Validation

- **Logic:** If a currency code is specified, it must exist in the currency code register.
- **File:** RA16L1 (Currency Codes)
- **Field:** A5VALK
- **Condition:** "The process will not allow a record if A5VALK does not exist in the currency code register."

2. Project Code Validation

- **Logic:** If a project code is specified, it must exist in the project register.
- **File:** RP1PL1 (Project Codes)
- **Field:** A5PROS
- **Condition:** "The process will not allow a record if A5PROS does not exist in the project register."

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 when selecting a firm.
- **Logic:** Retrieves firm details and sets user context.
- **Impact:** "This call initializes the firm context for the registration process."

2. RF011C (Label Printing Program)

- **Trigger:** Called when a transaction is registered.
- **Logic:** Generates labels for transactions.

- **Impact:** "This is a **destructive filtering step** that ensures labels are printed for registered transactions."

3. RS701R (Bilag Code Validation Program)

- **Trigger:** Called to validate the bilag code.
- **Logic:** Checks if the bilag code is valid according to the code register.
- **Impact:** "This call acts as a **major logical gateway** for ensuring valid bilag codes are used in transactions."