RF010R.MBR

Path: NXCLOUD/rpgsrc/RF010R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 15647ms

Business Logic for Bilagsregistrering

This document outlines the business rules that govern the process of bilagsregistrering (voucher registration), based on an analysis of the RPG program RF010R. The primary focus is on the rules and validations that ensure the integrity and correctness of financial entries.

The core logic for bilagsregistrering is contained within various subroutines in RF010R. The program handles the registration of vouchers, including validations for dates, amounts, and account codes, as well as interactions with subfiles for displaying and managing entries.

Order Status and Header Rules

Bilagsregistrering: RBUNL1, RTRANL1, RTRALE1, RAA1L1, RAA3L1, RLEVL1, RLEVRL1, RKUNL1, RKTRL3, RLTRL3

1. Valid Bilagsnummer Check

- **Logic:** The program checks if the bilagsnummer (voucher number) is unique for the current year and not previously used.
- **File:** RKTRL3 (Customer postings)
- **Field:** BILN
- **Condition:** The process will not allow registration if BILN already exists for the current year.

2. Date Validity Check

- **Logic:** Ensures that the bilagsdato (voucher date) is within valid limits (not in the future or too far in the past).
- **File:** RTRALE1 (Transaction records)
- **Field:** BILK
- **Condition:** The process will not accept a date that is greater than the current date or less than one year in the past.

Configuration and Authorization Rules

1. Account Type Validation

- **Logic:** Checks that both debit and credit accounts are not both receivable accounts.
- **Files:**
- RHOVL1 (Account details)
- RLEVL1 (Supplier details)
- **Fields:**
- KONT (Account)
- SPES (Special code)
- **Condition:** The process will not allow registration if both KONT fields are marked as receivable accounts.

2. User Authorization Check

- **Logic:** Validates that the user has permission to register vouchers.

•**File:** AA002R (User authorization)

•**Field:** USER

•**Condition:** The process will not proceed if the user does not have the necessary permissions.

Financial and Transactional Rules

1. Negative Amount Check

•**Logic:** Ensures that the amount being registered is not negative.

•**File:** RTRALE1 (Transaction records)

•**Fields:**

•BELØ (Amount)

•**Condition:** The process will not allow registration if BELØ is less than zero.

2. Currency Validation

•**Logic:** Checks that if a currency code is provided, it must be valid and correspond to the account's currency.

•**File:** RA505R (Currency codes)

•**Condition:** The process will not proceed if the currency code does not match the account's currency.

Special Conditions (Program-Specific)

1. Bunt Registration Check (RF010R)

•**Logic:** Validates that a buntnummer (bundle number) is either new or can be updated based on existing records.

•**File:** RBUNL1 (Bundle records)

•**Field:** BUNT

•**Condition:** The process will not allow registration if the buntnummer already exists for the current firm and year.

2. Interim Transactions Handling (RF010R)

•**Logic:** Allows for the registration of interim transactions with specific rules for handling amounts and accounts.

•**File:** RTRALE1 (Transaction records)

•**Fields:**

•BELØ (Amount)

•KONT (Account)

•**Condition:** The process will ensure that interim transactions are correctly flagged and processed according to business rules.

Subprogram Calls Affecting Logic

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

1. AA002R (User Authorization Check)

•**Trigger:** Called at the beginning of the program to validate user permissions.

•**Logic:** Checks if the user has the necessary permissions to perform the registration.

- **Impact:** This call acts as a **gatekeeper** for the registration process.

2. RF011C (Label Printing for MultiSupport)

- **Trigger:** Called when a voucher is registered and requires a label to be printed.
- **Logic:** Generates a label for the registered transaction.
- **Impact:** This is a **critical step** for ensuring that physical documentation is created for the transaction.

3. RS701R (Bilagskode Validation)

- **Trigger:** Called during the validation phase to check the validity of the bilagskode (voucher code).
- **Logic:** Validates the bilagskode against the code register.
- **Impact:** This is a **key validation step** that ensures only valid codes are used in transactions.

This documentation provides a comprehensive overview of the business logic implemented in the RF010R program for bilagsregistrering, detailing the rules, conditions, and subroutine interactions that govern the process.