AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 12:26:05 **Processing Time**: 12868ms

Business Logic for Number Series Calculation and Update

This document outlines the business rules that govern the calculation and updating of number series, based on an analysis of the RPG program AS100R. The primary focus is on how the program manages number series based on different input codes and conditions.

The core logic for number series management is contained within the various subroutines in AS100R. The program processes requests to retrieve, return, or validate number series based on parameters passed from calling programs.

Order Status and Header Rules

Number Series: NUMMER-REGISTER

1. Retrieve Standard Number

- **Logic:** When the input code (p_kode) is 0, the program retrieves the next available number from the automatic series.
- **File:** NUMMER-REGISTER (Stores number series details)
- **Field:** anuml1_key
- **Condition:** The program checks if the current number exceeds the maximum (anntom). If so, it wraps around to the starting number (annfom).

2. Return Last Used Number

- **Logic:** When the input code (p_kode) is 1, the program returns the last used number to the automatic series.
- **File:** NUMMER-REGISTER
- **Field:** ansist
- **Condition:** The last used number is only returned if it matches the current last used number.

Configuration and Authorization Rules

1. Manual Number Validation

- **Logic:** When the input code (p_kode) is 2, the program checks if a manually entered number is within the defined series limits.
- **Files:**
- NUMMER-REGISTER
- **Fields:**
- p_numm (the manually entered number)
- **Condition:** If the manually entered number is less than the starting number (annfom) or greater than the maximum number (anntom), an error code (p_kode = '7') is returned.

2. Check for Number Series Existence

- **Logic:** If no valid number series is found, the program sets the return code to indicate the absence of a number series.

- **File:** NUMMER-REGISTER
- **Field:** p_kode
- **Condition:** If the chain operation on the number series returns no records, the program sets p_kode to 9.

–

Financial and Transactional Rules

1. Update Number Series

- **Logic:** When retrieving the next available number, the program updates the number series in the database.
- **File:** NUMMER-REGISTER
- **Fields:**
- ansist (the last used number)
- **Condition:** The update only occurs if the chain operation successfully retrieves the record.

2. Wrap Around Logic

- **Logic:** If the next number exceeds the maximum defined number, the program checks if wrapping is enabled and adjusts the number accordingly.
- **File:** NUMMER-REGISTER
- **Condition:** If anwrap is set to 1, the next number wraps around to the starting number.

–

Special Conditions (Program-Specific)

1. Initialization Subroutine (AS100R)

- **Logic:** Initializes the program by setting up keys for the number register.
- **File:** NUMMER-REGISTER
- **Field:** anuml1_key
- **Condition:** The initialization occurs at the start of the program to prepare for number processing.

2. Parameter Handling (AS100R)

- **Logic:** The program receives parameters from the calling program to determine the operation to perform.
- **File:** NUMMER-REGISTER
- **Fields:** p_kode, p_firm, p_fell, p_type, p_fast, p_numm
- **Condition:** The parameters are evaluated to set up the necessary context for number series processing.

–

Subprogram Calls Affecting Logic

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

1. mannnr (Manual Number Check)

- **Trigger:** Called when validating a manually entered number.
- **Logic:** Checks if the manually entered number is within the defined series limits.
- **Impact:** If the number is out of bounds, it sets an error code that affects the overall process.

2. nyttnr (Next Number Calculation)

- **Trigger:** Called to calculate the next available number in the series.

•**Logic:** Increments the last used number and checks for wrapping conditions.

•**Impact:** This is a **critical step** in ensuring the integrity of the number series.

3. gammnr (Return Last Number)

•**Trigger:** Called when returning the last used number to the series.

•**Logic:** Validates that the number being returned is indeed the last used number.

•**Impact:** Ensures that the number series remains accurate and prevents duplication.