AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 12:22:47 **Processing Time**: 12180ms

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 the logic that determines how numbers are assigned, checked, and updated within the number series.

The core logic for number series management is contained within the various subroutines in AS100R. The program processes requests to retrieve the next available number, return a previously used number, and check if a manually entered number is valid within the defined range.

Order Status and Header Rules

Number Series Management: anuml1, anumlur

1. Retrieve Standard Number

- **Logic:** If the request code (p_kode) is 0, the program retrieves the next available number in the automatic series.
- **File:** anuml1 (Number Register)
- **Field:** anuml1_key
- **Condition:** The process will not select a record if the key does not exist in the number register.

2. Return Last Used Number

- **Logic:** If the request code (p_kode) is 1, the program returns the last used number to the automatic series.
- **File:** anumlur (Number Register)
- **Field:** ansist
- **Condition:** The last used number is only returned if it matches the current last used number in the system.

Configuration and Authorization Rules

1. Check Manual Number Validity

- **Logic:** If the request code (p_kode) is 2, the program checks if a manually entered number is within the defined range.
- **Files:**
- anuml1 (Number Register)
- **Fields:**
- p_numm (Manual Number)
- **Condition:** The manual number must be greater than or equal to annfom and less than or equal to anntom. If it is outside this range, an error code 7 is returned.

2. Update Number Register

- **Logic:** The program updates the number register with the next available number if the request code is 0.

- **File:** anumlur (Number Register)
- **Field:** ansist
- **Condition:** The update occurs only if the record is found in the number register.

Financial and Transactional Rules

1. Calculate Next Number

- **Logic:** The program calculates the next number to be used by incrementing the last used number (ansist). If the next number exceeds the maximum (anntom), it wraps around to the minimum (annfom).
- **File:** anumlur (Number Register)
- **Fields:**
- p_numm (Next Number)
- ansist (Last Used Number)
- **Condition:** If the next number exceeds anntom and wrapping is enabled (anwrap is 1), it resets to annfom.

2. Return Last Used Number Logic

- **Logic:** The program only allows the last used number to be returned if it matches the current last used number.
- **File:** anumlur (Number Register)
- **Condition:** If the number being returned is not the last used number, it remains unchanged.

Special Conditions (Program-Specific)

1. Initialization Subroutine (AS100R)

- **Logic:** Initializes the program by setting up the key fields for the number register.
- **File:** anuml1 (Number Register)
- **Field:** anuml1_key
- **Condition:** The initialization occurs when the program starts.

2. Parameter Handling (AS100R)

- **Logic:** The program accepts parameters from the calling program, including the request code and various identifiers.
- **File:** anuml1 (Number Register)
- **Fields:** p_kode, p_firm, p_fell, p_type, p_fast, p_numm
- **Condition:** The parameters are used to determine the operation to be performed on the number series.

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 the request code is 2.
- **Logic:** Validates if the manually entered number is within the defined range.
- **Impact:** This check ensures that only valid numbers are processed, preventing errors in number assignment.

2. nyttnr (Next Number Calculation)

- **Trigger:** Called when the request code is 0.
- **Logic:** Calculates the next available number in the series.
- **Impact:** This is a **critical step** in ensuring that the number series is updated correctly and that numbers are assigned in sequence.

3. gammnr (Return Last Used Number)

- **Trigger:** Called when the request code is 1.
- **Logic:** Determines the last used number and prepares it for return to the series.
- **Impact:** This process is essential for maintaining the integrity of the number series and ensuring that numbers can be reused appropriately.