AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 12:58:00 **Processing Time**: 15186ms

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 assignments based on different conditions and inputs.

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 specific input parameters.

Order Status and Header Rules

Number Series Management: NUMMER-REGISTER

1. Retrieve Standard Number

- **Logic:** If the input code (p_kode) is 0, the program retrieves the next available number from the automatic series.
- **File:** anumlur (Number Register)
- **Field:** ansist
- **Condition:** The program checks if the record is found; if not, it sets p_kode to 9 indicating the number series does not exist.

2. Return Last Used Number

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

Configuration and Authorization Rules

1. Manual Number Validation

- **Logic:** If the input code (p_kode) is 2, the program checks if the manually entered number is within the defined series range.
- **Files:**
- anumlur (Number Register)
- **Fields:**
- p_numm (Manual Number)
- **Condition:** The program sets p_kode to 7 if the number is outside the defined range (annfom to anntom).

2. Check Number Series Existence

- **Logic:** The program checks for the existence of the number series based on identification and type.

•**File:** anuml1r (Number Register)
•**Field:** anuml1_key
•**Condition:** If no record is found, it sets p_kode to 9.

Financial and Transactional Rules

1. Update Number Register

•**Logic:** If a new number is retrieved and is valid, the program updates the number register with the new number.
•**File:** anumlur (Number Register)
•**Fields:**
•ansist (Last Used Number)
•**Condition:** The update occurs only if the record is found.

2. Wrap Around Logic

•**Logic:** If the next number exceeds the maximum defined number (anntom), the program resets the number to the minimum (annfom) if wrapping is enabled.
•**File:** anumlur (Number Register)
•**Condition:** The program checks the anwrap flag to determine if wrapping is allowed.

Special Conditions (Program-Specific)

1. Initialization Routine (AS100R)

•**Logic:** The program initializes the keys for the number register and sets up the parameters received from the calling program.
•**File:** anuml1r (Number Register)
•**Field:** anuml1_key
•**Condition:** This routine is executed at the start to prepare the program for processing.

2. Calculate New Number (AS100R)

•**Logic:** The program calculates the next number to be used based on the last used number and checks for wrapping conditions.
•**File:** anumlur (Number Register)
•**Fields:** p_numm (Next Number), ansist (Last Used Number)
•**Condition:** The calculation is performed only if the last used number is valid.

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 input code is 2 to validate a manually entered number.
•**Logic:** Checks if the manually entered number is within the defined range.
•**Impact:** This call ensures that only valid manual numbers are accepted, preventing errors in number assignment.

2. nyttnr (Calculate Next Number)

•**Trigger:** Called when the input code is 0 to retrieve the next available number.

•**Logic:** Calculates the next number based on the last used number and checks for wrap-around conditions.
•**Impact:** This is a crucial step in ensuring that the number series is managed correctly and efficiently.

3. gammnr (Return Last Number)

•**Trigger:** Called when the input code is 1 to return the last used number to the series.
•**Logic:** Validates if the number being returned is the last used number and updates accordingly.
•**Impact:** This function helps maintain the integrity of the number series by ensuring that only the correct last number is returned.