AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 12:06:05 **Processing Time**: 12041ms

Business Logic for Nummerserie Management

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 number series are retrieved, updated, and validated within the system.

The core logic for number series management is contained within the *entry subroutine in AS100R . The program processes requests for number retrieval and updates based on specific codes sent from calling programs.

Order Status and Header Rules

Nummerserie Management: anuml1, anumlur

1. Retrieve Standard Number

•**Logic:** When the code sent is 0, the program retrieves the next available number from the automatic series.

•**File:** anumlur (Number Register)

•**Field:** ansist

•**Condition:** The process will not select a record if the number series is not found (*in66 is on).

2. Return Last Used Number

•**Logic:** When the code sent is 1, the program returns the last used number to the automatic series.

•**File:** anumlur (Number Register)

•**Field:** ansist

•**Condition:** The last used number can only be returned if it matches the current last used number.

Configuration and Authorization Rules

1. Manual Number Check

•**Logic:** When the code sent is 2, the program checks if a manually entered number falls within the defined series range.

•**Files:**

•anuml1 (Number Series)

•anumlur (Number Register)

•**Fields:**

•p_numm (Manual Number)

•anntom (Upper Limit of Series)

•**Condition:** The manual number must be greater than or equal to the starting number (annfom) and less than or equal to the ending number (anntom).

2. Update Number Register

•**Logic:** The program updates the number register with the next available number when the code is 0.

•**File:** anumlur (Number Register)
•**Field:** ansist
•**Condition:** The update occurs only if the record is found (*in66 is off).

Financial and Transactional Rules

1. Calculate Next Number
•**Logic:** The program calculates the next number to be used by incrementing the last used number.
•**File:** anumlur (Number Register)
•**Fields:**
•ansist (Last Used Number)
•p_numm (Next Number)
•**Condition:** If the next number exceeds the upper limit (anntom), it wraps around to the starting number (annfom).
2. Return Number Logic
•**Logic:** When returning a number, the program checks if the number being returned is the last used number.
•**File:** anumlur (Number Register)
•**Condition:** The number can only be returned if it matches the last used number; otherwise, it remains unchanged.

Special Conditions (Program-Specific)

1. Initialization (AS100R)
•**Logic:** Initializes the program by setting up keys for the number register.
•**File:** anuml1 (Number Series)
•**Field:** w_firm
•**Condition:** The program sets the firm number and other parameters before processing.
2. Manual Number Validation (AS100R)
•**Logic:** Validates if a manually entered number is within the defined series.
•**File:** anuml1 (Number Series)
•**Fields:** p_numm (Manual Number), anntom (Upper Limit)
•**Condition:** The manual number must be checked against the defined series limits.

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 code is 2 to validate a manually entered number.
•**Logic:** Checks if the manually entered number is within the valid 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 code is 0 to calculate the next available number.
•**Logic:** Increments the last used number and checks for wrapping conditions.
•**Impact:** This is a critical step for ensuring that the number series is managed correctly and does not exceed defined limits.

3. gammnr (Return Last Used Number)

•**Trigger:** Called when the code is 1 to return the last used number.
•**Logic:** Validates and adjusts the last used number if necessary.
•**Impact:** This represents a key function in managing the integrity of the number series, ensuring that previously used numbers can be reused appropriately.