AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 12:16:24 **Processing Time**: 11516ms

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 for various operations, including retrieving, updating, and validating numbers.

The core logic for number series management is contained within the main processing routine in AS100R. The program interacts with a number register to perform operations based on the input parameters provided.

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 will not proceed if the record is not found in the number register.

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 checks if the last used number is valid before returning it.

Configuration and Authorization Rules

1. Manual Number Check

- **Logic:** If the input code (p_kode) is 2, the program checks if the manually entered number is within the defined series range.
- **Files:**
- anuml1r (Number Register)
- **Fields:**
- p_numm (Manual Number)
- anntom (Upper Limit)
- annfom (Lower Limit)
- **Condition:** The process will return an error code (7) if the number is outside the defined range.

2. Automatic Number Update

- **Logic:** When retrieving the next number, if it exceeds the upper limit (anntom), it wraps around to the lower limit (annfom).

•**File:** anumlur (Number Register)
•**Field:** p_numm
•**Condition:** This logic is executed only when the input code is 0.

Financial and Transactional Rules

1. Update Number Register
•**Logic:** If a new number is retrieved, it updates the number register with the new value.
•**File:** anumlur (Number Register)
•**Fields:**
•p_numm (New Number)
•ansist (Last Used Number)
•**Condition:** The update occurs only if the record is found in the number register.
2. Return Error for Non-Existent Series
•**Logic:** If no number series is found during the check, the program sets the return code to 9 and the number to zero.
•**File:** anuml1r (Number Register)
•**Condition:** This occurs if all checks for identification fail.

Special Conditions (Program-Specific)

1. Initialization Routine (AS100R)
•**Logic:** Initializes the program and sets up the key fields for the number register.
•**File:** anuml1r (Number Register)
•**Field:** anuml1_key
•**Condition:** This routine is called at the start of the program to prepare for processing.
2. Calculate New Number (AS100R)
•**Logic:** Computes the next number to be used based on the last used number.
•**File:** anumlur (Number Register)
•**Fields:** p_numm (Next Number), ansist (Last Used Number)
•**Condition:** This calculation occurs only if the input code is 0.

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.
•**Logic:** Validates if the manually entered number is within the defined range.
•**Impact:** This acts as a **validation step** to ensure user input is acceptable.
2. nyttnr (Next Number Calculation)
•**Trigger:** Called when the input code is 0.
•**Logic:** Calculates the next available number in the series.
•**Impact:** This is a **critical step** in number series management, ensuring continuity in number assignment.

3. gammnr (Return Last Number)

- **Trigger:** Called when the input code is 1.
- **Logic:** Determines the last used number and prepares it for return.
- **Impact:** This represents the **restoration of the last used number**, maintaining the integrity of the number series.