AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 10181ms

---

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 operational codes.

The core logic for number series management is contained within the *entry subroutine in AS100R . The program processes requests to retrieve, return, or validate numbers from a number register, handling both automatic and manual number series.

Order Status and Header Rules

Number Series Management: NUMMER-REGISTER

1. Retrieve Standard Number

- **Logic:** If the request code (p_kode) is 0, the program retrieves the next available number from an automatic series.
- **File:** anumlur (Number Register)
- **Field:** ansist
- **Condition:** The program checks if the number retrieved is valid and updates the register accordingly.

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 program only returns the number if it matches the last used number.

Configuration and Authorization Rules

1. Manual Number Validation

- **Logic:** If the request code (p_kode) is 2, the program checks if the manually entered number falls within the defined range of the number series.
- **Files:**
- anuml1r (Number Register)
- **Fields:**
- p_numm (Input Number)
- **Condition:** The number is considered valid if it is greater than or equal to annfom and less than or equal to anntom.

2. Check for Number Series Existence

- **Logic:** If no number series is found during the lookup, the program sets the return code to indicate the series does not exist.

- **File:** anuml1r (Number Register)
- **Field:** *in66
- **Condition:** If the chain operation does not find a valid record, it sets p_kode to 9.

Financial and Transactional Rules

1. Update Number Register

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

2. Wrap Around Logic for Number Series

- **Logic:** If the next number exceeds the maximum defined (anntom), the program can wrap around to the minimum (annfom) based on the wrap condition.
- **File:** anumlur (Number Register)
- **Condition:** If anwrap is set to 1, the number will wrap around; otherwise, it sets p_kode to 8 indicating all numbers are used.

Special Conditions (Program-Specific)

1. Initialization of Program (AS100R)

- **Logic:** The program initializes by setting up keys for the number register and loading parameters from the calling program.
- **File:** anuml1r (Number Register)
- **Field:** anuml1_key
- **Condition:** The initialization occurs at the start of the program to prepare for number retrieval and updates.

2. Calculate New Number (AS100R)

- **Logic:** The program calculates 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:** The calculation adjusts for wrap-around conditions and updates the return code as necessary.

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 range.
- **Impact:** This affects the overall process by determining if a manual entry is valid or not.

2. nyttnr (New Number Calculation)

- **Trigger:** Called to calculate the next available number in the automatic series.
- **Logic:** Increments the last used number and checks for wrap-around.
- **Impact:** This is a critical step in ensuring that the number series remains valid and usable.

3. gammnr (Return Last Number)

- **Trigger:** Called to return the last used number to the series.
- **Logic:** Validates and adjusts the last used number based on conditions.
- **Impact:** This represents the handoff to the next major business function, ensuring that the number series is accurately maintained.