AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 15:03:35 **Processing Time**: 14418ms

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 retrieves, manages, and updates number series records based on various input parameters.

The core logic for number series management is contained within the *entry subroutine in AS100R . The program processes records from the number register files and performs actions based on the provided parameters.

Order Status and Header Rules

Number Series Management: anuml1, anumlu

1. Retrieve Standard Number

•**Logic:** The program retrieves the next available number from the automatic series if the input code (p_kode) is 0.

•**File:** anuml1 (Number Register)

•**Field:** anuml1_key

•**Condition:** The process will not select a record if *in66 is on, indicating that the record was not found.

2. Check Manual Number

•**Logic:** If the input code (p_kode) is 2, the program checks if the manually entered number is within the defined range.

•**File:** anuml1 (Number Register)

•**Field:** anuml1_key

•**Condition:** The process will return an error if the manually entered number is less than annfom or greater than anntom.

Configuration and Authorization Rules

1. Update Number Register

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

•**Files:**

•anuml1 (Number Register)

•anumlu (Number Update)

•**Fields:**

•ansist (Last used number from the register)

•p_numm (Next number to be assigned)

•**Condition:** The update occurs only if the record is found (*in66 is off).

2. Return Last Used Number

- **Logic:** If the input code (p_kode) is 1, the program attempts to return the last used number to the automatic series.
- **File:** anuml1 (Number Register)
- **Field:** p_numm
- **Condition:** The number can only be returned if it matches the last used number (ansist).

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:** anuml1 (Number Register)
- **Fields:**
- p_numm (Next number to be assigned)
- ansist (Last used number)
- **Condition:** If the calculated number exceeds the maximum (anntom), it checks if wrapping is allowed (anwrap).

2. Check Number Availability

- **Logic:** The program checks if all numbers in the series are used and returns an appropriate code if they are.
- **File:** anuml1 (Number Register)
- **Condition:** If the next number exceeds the maximum and wrapping is not allowed, it sets p_kode to 8.

Special Conditions (Program-Specific)

1. Manual Number Check (AS100R)

- **Logic:** The program checks if a manually entered number is valid within the defined range.
- **File:** anuml1 (Number Register)
- **Field:** p_numm
- **Condition:** The process will set p_kode to 7 if the manually entered number is outside the defined range.

2. Initialization Subroutine (AS100R)

- **Logic:** Initializes the program by setting up keys for the number register.
- **File:** anuml1 (Number Register)
- **Fields:**
- w_firm (Firm number)
- anuml1_fell (Field identifier)
- **Condition:** This subroutine is called at the start to prepare for processing.

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 that manual entries are correct.

2. nyttnr (Next Number Calculation)

•**Trigger:** Called when the input code is 0.

•**Logic:** Calculates the next available number in the automatic series.

•**Impact:** This is a **critical calculation step** that determines the number assignment.

3. gammnr (Return Last Used Number)

•**Trigger:** Called when the input code is 1.

•**Logic:** Handles the logic for returning the last used number to the series.

•**Impact:** This represents a **transactional adjustment** to the number series.