AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 13:32:02 **Processing Time**: 13553ms

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 the logic related to retrieving, updating, and validating number series within the system.

The core logic for number series management is contained within the *entry subroutine in AS100R . The program processes requests to retrieve the next available number, return a previously used number, or check if a manually entered number is valid.

Order Status and Header Rules

Number Series Management: NUMMER-REGISTER

1. Retrieve Standard Number

- **Logic:** The program retrieves the next available number from the automatic series based on the current state of the number series.
- **File:** NUMMER-REGISTER (Holds the number series data)
- **Field:** anuml1_key
- **Condition:** The process will not select a record if the number series is not found (indicated by *in66 being *off).

2. Return Last Used Number

- **Logic:** The program allows the last used number to be returned to the automatic series if it is the most recently used number.
- **File:** NUMMER-REGISTER (Holds the number series data)
- **Field:** ansist
- **Condition:** The return will only occur if p_numm matches ansist.

Configuration and Authorization Rules

1. Manual Number Check

- **Logic:** The program checks if a manually entered number is within the defined range of the number series.
- **Files:**
- NUMMER-REGISTER (Holds the number series data)
- **Fields:**
- p_numm (The manually entered number)
- **Condition:** The check fails if p_numm is less than annfom or greater than anntom, setting p_kode to '7'.

2. Automatic Number Retrieval

- **Logic:** The program calculates the next available number in the automatic series and updates the register.

•**File:** NUMMER-REGISTER (Holds the number series data)
•**Field:** ansist
•**Condition:** The next number is calculated based on the last used number, and if it exceeds the maximum, it wraps around or indicates all numbers are used.

Financial and Transactional Rules

1. Update Number Register
•**Logic:** The program updates the number register with the new number after retrieval.
•**File:** NUMMER-REGISTER (Holds the number series data)
•**Fields:**
•ansist (Last used number)
•**Condition:** The update will only occur if the record is found (indicated by *in66 being *off).
2. Wrap Around Logic
•**Logic:** If the next calculated number exceeds the maximum defined in the series, it wraps around to the starting number.
•**File:** NUMMER-REGISTER (Holds the number series data)
•**Condition:** This occurs when p_numm exceeds anntom and anwrap is set to '1'.

Special Conditions (Program-Specific)

1. Manual Number Handling (AS100R)
•**Logic:** The program handles specific conditions for manual number entries, ensuring they are checked against the series limits.
•**File:** NUMMER-REGISTER (Holds the number series data)
•**Field:** p_numm
•**Condition:** If the manual number is outside the defined range, it sets p_kode to '7'.
2. Initialization Subroutine (AS100R)
•**Logic:** The program initializes the parameters received from the calling program, setting up the keys for the number register.
•**File:** NUMMER-REGISTER (Holds the number series data)
•**Fields:** w_firm, anuml1_fell
•**Condition:** The initialization occurs at the start of the program to prepare for number 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 a manual number needs to be validated.
•**Logic:** Checks if the manually entered number is within the defined range of the number series.
•**Impact:** This acts as a **validation step** to ensure that manual entries are legitimate.
2. nyttnr (Next Number Calculation)
•**Trigger:** Called to calculate the next available number in the automatic series.
•**Logic:** Computes the next number based on the last used number and handles wrap-around logic.

•**Impact:** This represents a **critical calculation step** that determines the flow of number assignment.

3. gammnr (Return Last Used Number)

•**Trigger:** Invoked when the last used number needs to be returned to the series.

•**Logic:** Validates if the last used number can be returned based on the current state.

•**Impact:** This is a **conditional return step** that affects the availability of numbers in the series.