AS100R.MBR

Path: NXCLOUD/rpgsrc/AS100R.MBR **Generated**: 2026-01-08 15:42:59 **Processing Time**: 15321ms

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 handles number retrieval, validation, and updates based on various input parameters. 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 validate a manually entered number against defined ranges.

Number Retrieval and Validation Rules

Number Series Management: ANUMUL1, ANUMULU

1. Retrieve Next Available Number

- **Logic:** If the request code (p_kode) is '0', the program calculates the next available number in the automatic series.
- **File:** ANUMULU (Automatic Number Series)
- **Field:** p_numm
- **Condition:** The program will only retrieve a number if the current number does not exceed the defined maximum (anntom).

2. Return Last Used Number

- **Logic:** If the request code (p_kode) is '1', the program allows the last used number to be returned to the series.
- **File:** ANUMULU (Automatic Number Series)
- **Field:** p_numm
- **Condition:** The number can only be returned if it matches the last used number (ansist).

3. Manual Number Validation

- **Logic:** If the request code (p_kode) is '2', the program checks if the manually entered number (p_numm) falls within the defined range.
- **File:** ANUMUL1 (Manual Number Series)
- **Field:** p_numm
- **Condition:** The process will return an error code '7' if the number is outside the defined minimum (annfom) and maximum (anntom).

Update and Processing Rules

1. Update Number Series

- **Logic:** When retrieving the next available number, the program updates the number series with the new value.
- **File:** ANUMULU (Automatic Number Series)
- **Field:** ansist

•**Condition:** The update occurs only if the record is found and the request code is '0'.

2. Check for Number Series Existence

•**Logic:** The program checks if a number series exists based on the identification fields.

•**File:** ANUMUL1 (Manual Number Series)

•**Fields:** anuml1_fell, anuml1_type, anuml1_fast

•**Condition:** If no matching record is found, the program sets the return code (p_kode) to '9'.

Special Conditions (Program-Specific)

1. Wrap Around Logic

•**Logic:** If the next number exceeds the maximum and wrapping is allowed, the program resets the number to the minimum and increments it.

•**File:** ANUMULU (Automatic Number Series)

•**Field:** p_numm

•**Condition:** This logic is applied only if the wrapping flag (anwrap) is set to '1'.

2. Manual Number Handling (AS100R)

•**Logic:** The program checks if the manually entered number is the last used number before allowing it to be returned.

•**File:** ANUMULU (Automatic Number Series)

•**Field:** p_numm

•**Condition:** The return occurs only if the manually entered number matches the last used number (ansist).

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 request code is '2'.

•**Logic:** Validates if the manually entered number is within the defined range.

•**Impact:** This check prevents invalid numbers from being processed further.

2. nyttnr (Next Number Calculation)

•**Trigger:** Called when the request code is '0'.

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

•**Impact:** This calculation is crucial for ensuring that the number series remains sequential and valid.

3. gammnr (Return Last Number)

•**Trigger:** Called when the request code is '1'.

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

•**Impact:** This ensures that the number series can be reused correctly without duplication.

This documentation provides a comprehensive overview of the business logic encapsulated within the AS100R RPG program, detailing how it manages number series through various operations and checks.